

Commodore INFOC

PRIJS f 7.25/Bfr. 135

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS

JAARGANG 4, NO. 7, NOV./DEC. 1987

LISTINGS

Solitaire
Powerboat
Tekstverfraaien
Duo-bal
Explosieven
Files uitprinten
Springbeest
Vaste hand C-128
Mastermind C-128
Mt-Base C-16
Adresbestand C-16

Fresh
HiFi-schakelingen
Desktop Video
MS-DOS op de Amiga
Amiga Genlocker

Vaste rubrieken
Basic Miniatuurjes
Machinetaal
Basic cursus
Oud van Goudriaan
Listings

Commodore-Info

Verschijnt 8 x per jaar

Jaargang 4, no. 7 nov/dec 1987

Uitgave:

Sala Communications

Uitgever:

Drs. J. Taverne

Redactie:

Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdredacteur

J. Boers, drs. M. de Rooij, drs. H.
Zoete, P. Boncz, B. Munniksma,
K. v.d. Vlies, drs. U. Schuurmans,

Redactiesecretariaat:

R. van Zalingen

Strip:

Bert Tier

Illustraties:

Advertentie-exploitatie:

Ing. V. Sala
Den Texstraat 5a
1017 XW Amsterdam
tel.: 020-273198

Redactie adres:

Postbus 112
1269 AC Blaricum
tel.: 02152-65695

Abonnementen en administratie:

Postbus 5570
1007 AN Amsterdam
tel.: 020-273198

Vragen betreffende abonnementen ont-
vangen wij bij voorkeur schriftelijk, met
meesturen van het omslagetiket.

Abonnement:

Voor 8 nummers f 47,50 of BFR. 950
per jaar. Betaling op Giro 1585491 t.n.v.
SAC/COMMODORE-Info.

Oude nummers kunt U alleen krijgen bij
vooruitbetaling van f 6,75 op de boven-
staande rekening.

Ook telefonische opgave voor een
abonnement is mogelijk. Bel GRATIS
06-02242222 (Teleservice), elke dag
tot 20.30 uur (dus ook in het weekend).
België: 115555, dagelijks tot 22.00 uur.
Deze telefoonnummers zijn alleen be-
doeld voor telefonische opgave van
NIEUWE abonnementen.

Druk:

Verweij, Mijdrecht
NDB, Zoeterwoude

Distributie:

In Nederland: Betapress/Gilze
In België: AMP/Brussel

© 1987 COMMODORE INFO

Alle rechten voorbehouden
ISSN: 0169-3085

Inhoud van dit nummer

Genlocker A8702

6

De nieuwe Amiga Genlocker is uit!

Nieuws

10, 13

Fresh

12

Een software-pakket voor huisartsen, dat
zowel de verrichtingen als de financiële af-
handelen daarvan verzorgt.

Software

15

Er zijn mensen, die alleen al voor de soft-
ware bespreking dit blad kopen.: Drie pa-
gina's met de nieuwste pakketten.

Strip

18

Bert Tier laat zien dat het in de jungle ook
niet altijd koek en ei is.

Basic Basic 16

19

Jan Bodzinga's vaste Basic-cursus, waar-
door het werken met de Commodore-ma-
chines veel meer mogelijkheden biedt.

Goud van Oud

27

We hebben hem een keer moeten missen,
maar hier is Rob Goudriaan weer met ga-
mes uit de oude doos.

Desktop Video

55

Ongekende mogelijkheden in beeldmani-
pulatie wordt door Ulco Schuurmans na-
der bekeken en aan u uitgelegd.

Minigraph (slot)

59

Het slot van de minigraph-listing,

Reset

62

Een eenvoudig, maar handig commando
om uw werkgeheugen schoon te krijgen.

Redactioneel

Nu de beurzenstorm van dit najaar
wat is geluwd en de rook is opgetrok-
ken, lijkt het weer even tijd voor
andere zaken. Dat wil niet zeggen,
dat er dit jaar geen nieuws meer te
verwachten is, want in de VS zijn op
dit moment een groot aantal beurzen
aan de gang.

In ons land maakt men zich intussen
op voor de jaarlijkse HCC- dagen en
aansluitend komen de feestdagen
snel dichterbij. Maar voor het zover is
een najaarsnummer, boordevol nieu-
we programma's en (software)-be-
sprekingen.

De trouwe lezers zal het zeker opval-
len, dat de Listing-rubriek in een
nieuw jasje is gestoken. Dank zij de

Datakolom

64

Luc Sala belicht het ontstaan van een 'te-
minologie'-cultuur in computerland.

HiFi-schakelingen

65

Een artikel over moeilijkheden, die de com-
puter kan bieden bij het schakelen van
audio-apparatuur.

Demowereld

66

Darius Heydrapour met het laatste nieuws
uit de wereld van computerkrakers.

Vragen

68

Onze vaste rubriek met vragen van ge-
bruikers op elk terrein.

Miniatuurtjes

72

Peter broekhuizen levert weer twee blad-
zijden met alleraardigste mini-listings.

Machinetaal 9

75

Voor de echte programmeurs de negende
aflevering van de cursus machinetaal.

MS-DOS

79

Het werken met en in MS-DOS op de Ami-
ga kan de nodige moeilijkheden opleve-
ren. Maar ook veel plezier!

Print-out

33

Listing-rubriek met::

Slitair	34
Powerboat	36
Tekstverfraaien	36
Duo-bal	38
Explosieven	40
Files uitprinten	40
Springbeest	45
Vaste hand C-128	46
Mastermind C-128	47
Mt-Base C-16	49
Adresbestand C-16	51

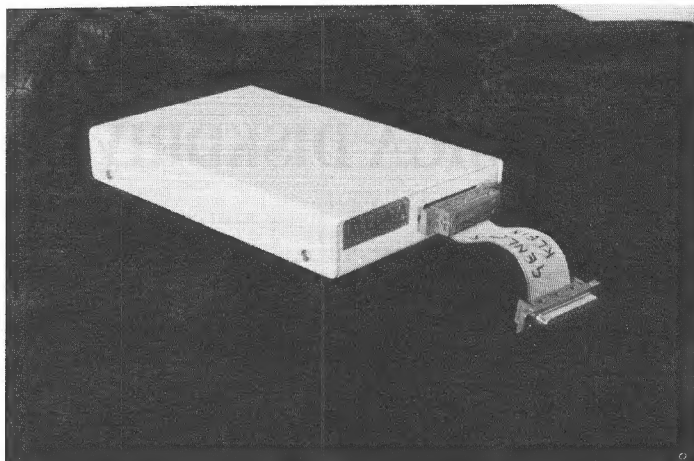
technische mogelijkheden op de re-
dactie kunnen deze pagina's nu
rechtstreeks van de C-64 in onze ru-
briek worden ingevoerd. Er zullen
nog wat vormtechnische schoon-
heidsfoutjes inzitten, maar die zullen
ongetwijfeld de volgende keer verhol-
pen zijn. Maar als het goed is zal de
'Missers'-rubriek niet meer nodig blij-
ken te zijn.

Voor de Amiga-bezitter zal ook uit dit
nummer weer duidelijk worden welke
tot voor kort ongekende mogelijkhe-
den deze Commodore-machines bie-
den. En wij kunnen u verzekeren, dat
het eind van die ontwikkeling voor-
alsnog niet in zicht is.

MdR

Met een genlock interface is het mogelijk om grafische computerbeelden en titels met videopnamen te combineren. Tot op heden waren genlockers voor de Amiga helaas schreeuwend duur of van slechte kwaliteit. Alldata brengt daar nu met haar nieuwe genlock interface verandering in.

Amiga Genlock A8702



**Professionele kwaliteit
ook voor de amateur?**

De Amiga 500, 1000 en 2000 staan in het middelpunt van de groeiende belangstelling voor desktop video (DTV). De grafische prestaties en animaties van deze machines zijn beroemd. Alleen konden de amateur en vakman met een krap budget daar niet veel mee.

De randapparatuur voor het genereren van een kwalitatief goed PAL-sig-naal en het combineren van computer graphics met videobeelden was gewoon te duur. Aan geschikte genlockers kon men ook in het buitenland nauwelijks komen en goedkope exemplaren bleken dikwijls beneden de maat. Daarom komt het minder dan f 1000,- kostende genlock interface van Alldata als geroepen. Een reden voor ons om deze genlocker eens nader te bekijken.

Beeldmixen

Genlockers combineren zoals gezegd computervideo- met bestaande of life opgenomen videobeelden.

Men spreekt ook wel van beeldmixen. Bekend zijn de titelletters die over de videobeelden heenlopen. In vele televisieseries komen de letters gewoon als titelrol op de videobeelden. Een techniek die ook met sommige videocamera's mogelijk is.

Het gebruik van een computer als af-titelaar biedt veel meer mogelijkheden dan een gewone titelgenerator voor video. Niet alleen kan de videomaker kiezen uit meerdere fonts in vrijwel

elke bruikbare grootte en kleur, ook kan men de titels laten scrollen, roteren om een X-, Y- of Z-as en van vorm laten veranderen.

Behalve titels kunnen ook grafieken, geanimeerde poppetjes en met paintpakketten ontworpen achtergronden met life/bestaande videobeelden gekombineerd worden. Een geliefde techniek bij reclamebureaus, interactief onderwijs en grappige films.

Hoe werkt een Genlock?

Een genlock kent doorgaans twee video-ingangen en een of meer video-uitgangen. Op de video-in worden twee verschillende videobronnen aangesloten. Bij de **A8702** zijn dat de Amiga via de 23-polige RGBD-connector en een composiet-videoingang voor een videocamera, videorecorder, beeldplaat, een andere computer, een satelliet-signaal of een professionele "black and burst". De video timing-signalen voor line, frame en colour burst van de Amiga en de externe videobron worden door het interface **ge-locked**. Dat wil zeggen zodanig op elkaar afgestemd dat beide signalen dezelfde kleur op dezelfde tijd en

plaats op het scherm (of naar de videorecorder) geven. Men spreekt hierbij ook wel van het synchroniseren van twee videobronnen. Zijn de beide videosignalen eenmaal gelocked dan kan de electronica tussen het Amiga-en externe videosignaal switchen.

Daarvoor gebruikt de genlocker een high speed videoschakelaar. Met behulp van deze schakelaar is het mogelijk om videobeelden als het ware over elkaar heen (**overlay**) te leggen waardoor gemixde videobeelden ontstaan.

Behalve het eigenlijke locken vervult de Amiga genlock A8702 nog twee andere belangrijke taken:

- **Encoding**, het omzetten van het Amiga **RGB**-signaal in een voor de monitor en/of videorecorder bruikbaar **composiet PAL-sig-naal**. Voor het functioneren van de encoder is een PAL-referentie-sig-naal nodig om de computersignalen opnieuw te timen.
- **Video overlay switching**, het schakelen tussen het computer- en het externe videosignaal. Er zijn vier schakelmogelijkheden: Amiga Display, display externe videobron

(default), de background- en foreground-modus. In de **background-modus** wordt de achtergrondkleur doorzichtig en het externe videobeeld (CVBS-sigitaal) zichtbaar in mixage met de computer graphics, bijvoorbeeld een aftiteling. Het register 0 wordt hier als de schakelaar gebruikt. In de **foreground-modus** is zwart de doorzichtige kleur. Daarmee kan een (voor- en achtergrond-)kleur transparant gemaakt worden om een extern videobeeld in een venster te mixen.

Voor de vakman/vrouw is er de mogelijkheid om Black and Burst, het aanbieden van een effen (zwarte) achtergrond voor overlays te gebruiken. De amateur zal in de meeste gevallen een normaal composiet videosignaal voor de overlay gebruiken.

De A8702

De door Alldata geïmporteerde A8702 genlock beschikt over alle hiervoor genoemde mogelijkheden en heeft daarbij nog twee verschillende uitgangen:

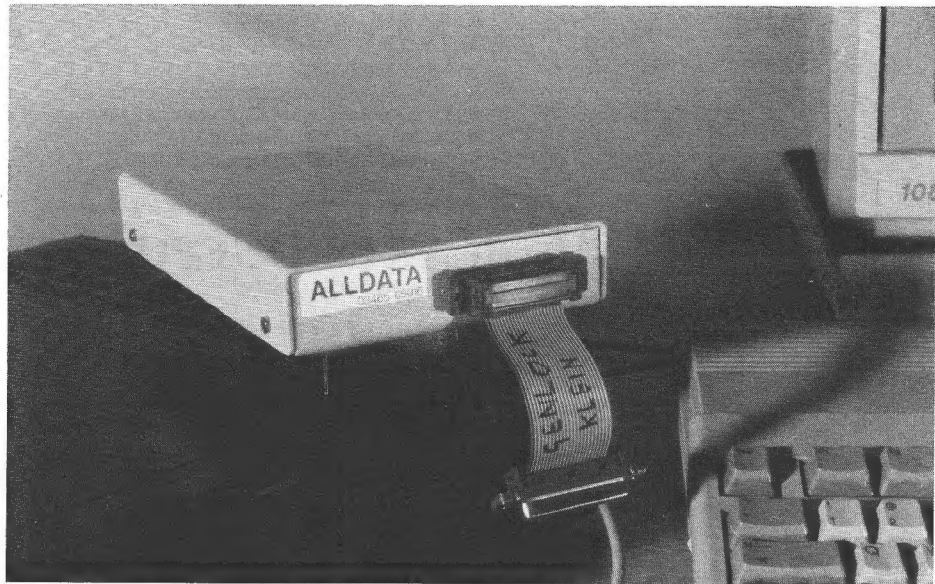
- ° Een gewone **composiet BNC**-uitgang voor het aansluiten van een videorecorder of monitor. Uit deze output komt het gecombineerde (overlay) PAL-sigitaal.
- ° De **23-polig RGB Amiga D-connector** voor: Rechtstreekse aansluiting van de eigen Amiga monitor. Twee optionele schakelaars voor het switchen tussen Amiga-beeld, gemengd beeld, videobeeld op de voorgrond en Amiga-beeld op de voorgrond. En softwarematige aansturing via de parallelle poort van de Amiga.

Het interface, in ons geval een testmodel, is een net plat metalen kastje ter grootte van een pocketboek. Uit dit kastje komt een lintkabeltje voor aansluiting op de Amiga D-connector. Dit kabeltje bleek voor onze Amiga 1000 wat aan de korte kant om het interface gemakkelijk bereikbaar naast de machine te zetten. Met enig gefriemel past het kastje gedeeltelijk onder de Amiga en bleven de beide schakelknoppen toch binnen bereik.

Alldata heeft toegezegd dat voor alle Amiga's eigen aangepaste uitvoeringen komen, zodat de gebruiker geen ongemak meer zal ondervinden.

Pluggen

Verder zitten er zoals gezegd op de genlocker een BNC video-in, -out en 23-polige D-connector. Twee, nog ongelabelde, drukknoppen regelden de



Amiga's nieuwste Genlock A8702

verschillende modussen van de A8702.

Het aansluiten van de genlock geeft geen probleem, zolang men de nodige verloopjes in huis heeft. Alldata levert de lintkabel er natuurlijk bij. Deze is van een afwijkend TTL-type. Zelf er eentje solderen kan tot het opblazen van de RGB-poort leiden. Voor de videokabeltje moet u zelf zorgen. Videorecorders gebruiken tulp-, SCART en BNC-pluggen. Voor tulp- en Scart zijn verloopstekkers en kabels bij een goede electro-nicazaak te koop. Neem wel enkele tientjes mee want goedkoop zijn deze supplies beslist niet! Desgewenst kan de gebruiker de kabeltjes met opgave van de juiste aansluitingen ook bij Alldata bestellen.

De videopraktijk

Eerlijk gezegd waren wij behoorlijk sceptisch toen Alldata contact opnam om hun genlocker te laten testen. Ervaringen uit het verleden met slechte of peperdure genlockers waren nu niet bepaald gelukkig geweest en ook vele lezers klaagden hierover. Al bij de eerste demonstratie verdwenen de meeste vooroordelen opslag. Het is allemaal nog niet op en top broadcasting-kwaliteit, maar de A8702 komt daar wel heel dicht bij. Niet alleen op de monitor, maar ook op de videorecorder. Met name bij de VCR onderscheidt de kwaliteit van deze genlocker zich van andere goedkope apparaten.

Het maken van overlays en het switchen tussen de verschillende modes kan iedereen in vijf minuten leren. Genlocker aansluiten op de Amiga, externe videobron en output, software

(kickstart en video- of paintpakket) in de Amiga en de voorstelling kan beginnen. Grote creatieve mogelijkheden bij een goede videokwaliteit. Volgens Alldata zit er nog een **enhancer** voor het wegwerken van de onvermijdelijke kopieerverliezen in de ontwerpkoker. Daar wilde men op het moment van schrijven verder nog niets over zeggen.

Na het opstarten is de default videobeelden op de achtergrond en computer-graphics op de voorgrond. De kickstart-ikoon wandelt eerst door het beeld heen. Dat verdwijnt na het inladen van de werkbank of video/paintpakket. Voor het veranderen van de videomodus drukt de gebruiker op de desbetreffende schakelknop van het interface. Ons inziens verdient het aanbeveling deze schakel-optie er gewoon bij te kopen gezien het grote gebruiksgemak dat men daar van heeft. Het effect van fore- en background-modus is afhankelijk van de gebruikte kleurinstellingen. In de background-modus krijgt de gebruiker de beste resultaten als er ook een zwarte achtergrondkleur gekozen wordt. Dat voorkomt het doorlopen van de kleuren. In de foreground-modus, waar de Amiga graphics transparant zijn, verdient het aanbeveling om de preferences naar smaak in te stellen. Aanbevolen wordt om Rood en Groen minder dan 50% in te stellen en blauw op full te zetten. Hier door ontstaat een professioneel blauw overlay-venster. Gebruik nimmer dezelfde voor- en achtergrondkleur, kleurkleurkleurkleur want dan wordt de achtergrond niet doorzichtig. Register 0 is dan inactief.

Lastig

Een klein probleempje geeft het encoderen van het Amiga RGB-signaal. Zonder een extern referentiesignaal komt er niet veel bijzonders uit de video-out. De gebruiker zal de genlocker dus een tuner- of anderszins moeten aanbieden voor het in fraaie PAL-kleuren vertonen van de Amiga-graphics. Dan behoeft de A8702 niet "droog te zwemmen". Een alternatieve mogelijkheid is het aanbieden van een zwartwit-sigitaal voor het creëren van zwartwit-overlays.

Opstelling

De video-configuratie is uiteraard afhankelijk van het beoogde doel. Videomakers zullen de Amiga voor met genlocker gebruiken voor mixages en aftiteling. Daarvoor zijn nodig een videocamera of videorecorder voor weergave, een opnamerecorder en/of output-monitor. Bij voorkeur maakt men gebruik van meerdere monitoren: een voor de Amiga zelf en een voor het controleren van het composiet outputsignaal. Let er bij het gebruik van een monitor op dat deze op AV/VCR staat anders wordt de output niet korrekt weergegeven.

Bij interactief onderwijs maakt men gebruik van een door de Amiga gestuurde beeldplaat- of videorecorder. Het interactieve Amiga-programma stelt de vragen of verschaft de begeleidende teksten. De externe videobron levert de bijbehorende beelden.

Gebruiksmogelijkheden

De toepassingen van de Amiga met genlock zijn legio. Behalve videoproducties, reclamerepresentaties, slideshows en interactief onderwijs zijn er nog tal van andere gebruiksmogelijkheden. Alldata noemt zelf trainings-systemen, het gelijktijdig bekijken van twee videobronnen (bijvoorbeeld een computer- en een TV-programma), informatie-terminal en toepassing als een video-database.

Conclusies en opmerkingen

De A8702 Genlock is een welkome aanvulling voor de videoproducent of onderwijskundige die met de Amiga werkt. Voor f 831,67 (exclusief B.T.W.) krijgt de koper een kwalitatief goede genlocker met tal van krachtige en creatieve mogelijkheden. Verwacht echter geen wonderen van deze genlocker. Het geringe kwali-

teitsverlies (high frequency details) van het omzetten van RGB naar PAL-composiet blijft zichtbaar. Hetzelfde geldt bij opname met videorecorders en vertoning op de huiskamer-TV. Deze achteruitgang ligt niet aan de genlocker, maar is inherent aan de kwaliteit van de overige randapparatuur.

De A8702 kan het signaal van een VCR goed tracken zolang dit van een redelijke kwaliteit is. Een matige bandkwaliteit en versleten of vuile recorderkoppen geven een slecht resultaat. Hetzelfde geldt voor videotraces waarbij niet-standaard synchronisatiesignalen gebruikt worden. Afgezien van de hier gemaakte kanttekeningen biedt de A8702 veel waar voor zijn geld. In zijn prijsklasse kent deze genlocker in Nederland momenteel geen concurrentie. Wie meer wil zal ook veel meer moeten betalen. En als de beloofde enhancer gereed is zal ook de verwende gebruiker van consumenten-video geheel aan zijn/haar trekken komen.

Voor alle info: Alldata, , tel. : 03465-65016/73767.

U.S.

NIEUW



SPECIALE AANBIEDINGEN:

Eprommer II No:65

f 139.50

Epromprogrammer voor de verschillende type eeproms 2716 t/m 27256. Met de Epromprogrammer kunt u ook de A en de Cmos eeproms programmeren. (verschillende programmeer spanningen 12.5/21/25 volt.) De epromprogrammer is helemaal softwarematig gestuurd. (Nederlandse software Nederlandse beschrijving.). Men kan met de Epromprogrammer eeproms uitlezen, vergelijken, kopiëren en natuurlijk programmeren. Verder zit er in de software een moduulgenerator die de programma's, omschrijft voor eprom (Tot 16K). Men kan zo zijn eigen programma's of die van andere op eprom zetten. Zo kan men eenvoudig meerder programma's in 1 eprom plaatsen die door middel van een menu eenvoudig zijn aan te roepen.

Superepromkaart No:832

F 135.00

Deze Superepromkaart is een ideaal hulpstuk voor mensen die vaak dezelfde programma's gebruiken. (BV.Superbase, Vizawrite, Copieerprogrammas enz.). Men kan dan namelijk snel en zonder te zoeken een programma gebruiken. Men kan op deze Superepromkaart bijna 2 volle diskette plaatsen. Het omschrijven van de programma's voor eprom regelt de kaart zelf door zijn unieke moduulgenerator. Deze kan 1 delige programma's ten grote van maximaal 172 blocks (40K) omschrijven. Op de Superepromkaart kan men 2764 t/m 27256 eeproms plaatsen. Het programmeren van deze eeproms dient men met een epromprogrammer te verrichten.

NIEUW

Epromkaartje No:220

f 19.50

Dit is een leuk klein epromkaartje met vele mogelijkheden. Men kan hier 2 8k of 2 16k eeproms opplaatsen zonder zelf iets te solderen of teverbreken. Op het kaartje is verder een mogelijkheid om een resetknop te plaatsen.

Universele print No:210

f 19.50

Een kaartje voor de hobbyisten. Men kan dit epromkaartje voor verschillende doeleinden gebruiken. Als moduul of externe kernal kaart. Wanneer men het Universelekaartje als moduulkaartje wilt gebruiken dan heeft men de keus, deze op verschillende wijze te bestukken. Als een 8k kaartje of als een 16k Kaartje (Beide mogelijkheden zijn softwarematig uitschakelbaar.) De Universele print past precies in een moduulkastje (No:1270) Verder is er nog een mogelijkheid om er een resetknopje op teplaatsen. (Compleet bestukt als 16K zijnde No:211)

COMMODORE ICS

IC Type	Type Comp/Drive	Prijs.	IC Type	Type Comp/Drive	Prijs.
6510	C64	F 29.95	6502	1541	F 19.50
6526	C64	- 29.95	6522	1541	- 19.50
6569	C64	- 84.00	901229	1541	- 42.00
6581	C64	- 65.00	325302	1541	- 48.00
901225	C64	- 37.50	325572	1541	- 48.00
901226	C64	- 40.00			
901227	C64	- 40.00	8502	C128	- 52.50
906114	C64	- 29.95	8701	C128	- 58.00
8701	C64	- 58.00	8721	C128	- 58.00
			8722	C128	- 58.00
8501	C16	- 42.50	8563 R9	C128	- 135.00
8360	C16	- 82.00	8566 R3	C128	- 92.50

Nieuw binnengekomen AMIGA IC 8520 uit voorraad leverbaar
F 107.50

Op de C-128 kan een tamelijk brede administratie een plaatsje vinden. Datahome maakte een financieel pakket voor huisartsen, dat ook de patientenregistratie afhandelt.

Fresh

Financieel registratiesysteem huisartsen

De C-128 is met zijn wat grotere geheugen en professionelere toetsenbord (bij de D) ingebouwde diskdrive heel bruikbaar voor administratiewerk. Voor de huisartsenpraktijk is er dus nu ook een pakket afgestemd op deze Commodore.

Het bedrijf Datahome is nogal actief op het Commodore softwarefront en heeft in feite de plaats ingenomen van heel wat grote importeurs en softwarehuizen uit het recente C-64 verleden.

Met een C-128 D, een kleurenmonitor, een Star NL-10 en het software-pakket Fresh bieden ze nu de huisarts een relatief goedkope oplossing. Voor het totaal betaalt men inclusief de hardware f 3.798,- incl. BTW. en dan komt men het pakket ook nog aan huis afleveren, installeren en demonstren. De software is ook afzonderlijk verkrijgbaar, maar dan met wat minder "after-sales" service, het kost in dat geval f 399,-. Er zit wel garantie op Fresh, maar liefst 5 jaar, maar zolang zal uw C-128 het vast niet volhouden, de ontwikkeling gaat daarvoor te snel.

Registratie verrichtingen

Zoals meestal het geval is bij een administratiepakket ligt aan Fresh een database ten grondslag. Daarin zitten de patienten, hun gegevens en daar kan dan de verrichting of behandeling aan worden toegevoegd. Met het Fresh pakket kan de huisarts of zijn assistente dan registreren wat er gebeurt, maar bijvoorbeeld ook de rekening in de vorm van nota's en een acceptgirokaart aanmaken.

Hoofdmenu

Het pakket is, en zo hoort dat ook voor dergelijke "verticale" software waarbij

de gebruiker niet veel van de computer behoeft te weten, helemaal menu-gestuurd. In het hoofdmenu kan men kiezen uit een aantal algemene procedures, uit het ingeven van de standaard verrichtingen, het printen van de nota's (facturen), het bijwerken van betaalde nota's, het maken van overzichten en aanmaningen en natuurlijk de registratie van de patienten. Dat laatste vormt het hart van het systeem, en de gegevens per client/patient of relatie kunnen in een aantal overzichten worden weergegeven.

Algemene opzet

De database in Fresh kan maximaal 1250 relaties (patienten) bevatten en dat is misschien iets aan de krappe kant voor een normale praktijk. Maar wanneer men de hele zaak alleen voor particuliere patienten gebruikt,

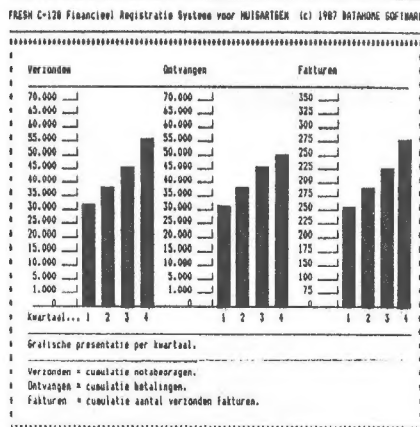
en die krijgen tenslotte de rekeningen, is het wel voldoende groot.

Er kunnen 28 soorten verrichtingen met codes worden vastgelegd, met een maximum van 2000 verrichtingen per boekingsperiode (kwartaal). Na het sturen van de acceptgiro's kunnen ook de betalingen natuurlijk verwerkt worden, ook als er sprake is van deelbetalingen. Wie niet betaalt, krijgt een herinnering of een kopie van de nota. Er kan ook een set stickers worden gemaakt, wanneer bijvoorbeeld een verhuisbericht verstuurd moet worden.

Voor de boekhouder zijn natuurlijk kwartaaloverzichten te maken, maar ook is de "omzet" in grafiekvorm te bestuderen.

Alles bij elkaar is dit systeem een vrij diepgaande aanpassing van een standaard computeradministratie aan de speciale wensen van het medische beroep. Er zullen zeker opmerkingen gemaakt worden, dat een C-128 toch niet groot genoeg is voor zoiets, maar wie al zo'n apparaat heeft of wie eens voorzichtig wil beginnen aan wat automatisering op bescheiden schaal in zijn praktijk, kan hier heel aardig mee uit de voeten. Het is geen complete vervanging van de administratie, het betreft namelijk alleen de uit te sturen nota's, inkomende rekeningen moeten op een andere manier worden bijgehouden, maar het is zeker een goed doordacht en aangepast stukje software.

Inl. Datahome, 020-855204.



Catch 23

Van Martech komt dit militair aandoend adventure, dat tamelijk ingewikkeld is. De speler komt terecht op een eiland, op n van vijf lokaties, ieder in de buurt van een "safe" huis. Het doel is natuurlijk om het er weer levend af te brengen, alle opdrachten te vervullen en ondertussen punten te verzamelen. Het tot nu toe behaalde maximum is 83700 punten, en dat in een tijd van 2 uur en 38 minuten, maar de gemiddelde avonturier zal er heel wat langer over doen. Het spel is af te breken en opnieuw te starten, indien men dat in een "safe" huis doet. Daarvoor moet de "timer" op 00:00 gezet worden.

Er zijn 6 zones, ieder vol verdachte huizen met "transmitter"-kamers, gevaren, tanks, bommen en patrouilles, maar middels een "dangerometer" kan men de situatie beoordelen. Als extra heeft men een mijndetector, die echter maar 15 minuten functioneert, tenzij men wat batterijen weet op te sporen in de gebouwen op het eiland. De oplossingen voor de adventure kunnen zoals gebruikelijk gevonden worden door goed te letten op de aanwijzingen, een voorwerp op een tafel kan al een aanwijzing bevatten over wie een bepaalde "transmitter" kamer bewoont. Wanneer die goed geraden wordt, kan men gaan zenden, natuurlijk weer met de nodige codes om bijvoorbeeld de CK23 "interceptor" van brandstof te voorzien.

De verschillende transmitters moeten allemaal gevonden worden om verder te komen.

Op het einde van het spel komt men in de 6e zone, die erg gevaarlijk is, maar wel de laatste transmitter en de CK23 interceptor bevat.

Alles bij elkaar een hele zoekpartij, een middenklasse adventure voor de C-64 waardig.

Zodiac

Search for the secret of life

Van Incentive software uit Berkshire UK, kregen we dit pakket opgestuurd en bij alle serieuze adventures moesten we hier af en toe wel om lachen. Humor kan de makers niet ontzegd worden, al is dit "graphic" adventure zeker geen grapje. In de drie delen ervan komt men, bij het verzamelen van de tekens van de dierenriem, in aanraking met diepe problemen, afgezien nog van de tegenstander Ramus. men heeft dit pakketje (op cassette)

de "double gold" aanduiding gegeven en het is inderdaad een uitdagende en toch speelbare adventure (in drie delen) met een wat diepgaander inslag dan de gebruikelijke zoek en meeneem tactieken. Via passwords krijgt men steeds toegang tot het volgende deel, dus men hoeft niet steeds de hele zaak te doorlopen, als men eenmaal door een deel is heengegaan.

Op de keerzijde van de cassette staat



nog een tweede adventure, "op zoek naar het geheim van het leven" en ook dat is wel een aardig gegeven, al is het zoeken in twee huizen/kastelen misschien wat gewoontjes voor de ware avonturier.

Ninja Hamster

Vroeger noemde men dit karate spelletjes, nu is Ninja in de mode. Toch is onze held in dit spel wel weer wat anders dan de gebruikelijke trappende en slaande oosterling. Het is namelijk een knaagdier, de Ninja hamster. Hij heeft een nobel doel, namelijk zijn dorp te bevrijden van slechte vijanden, wanneer hij terugkomt van een reis. Die vijanden hebben de vorm van een rat, een hagedis en meer van dat ongedierte. En dan is het vechten gebazen, met trappen, bijten en slaan. Op disk gaat dit C-64 spel ongeveer 60 gulden kosten.

I Alien

Van het Engelse CRL komen er een paar interessante spellen, zij het dat de thema's wat afgekloven zijn. Van de zoveelste vechtgame met een Ninja als hoofdfiguur is niemand echt onder de indruk. I Alien is voor de C-64 en gaat over een arme held, die bedreigd wordt door mensen van de planeet Aarde, die hem willen meenemen als proefkonijn voor experimen-

ten. Maar de "Alien" heeft een kans om te ontsnappen, wanneer hij zijn ruimteschip weet te bereiken. Maar dat is nu juist de essentie van dit verder niet opzienbarende spel, op weg daarheen vliegen de vijanden in allerlei gedaantes weer in het rond. De ijsdraken, droids, mutanten, kortom de hele batterij ellende moet overwonnen worden. En wie daar doorheen komt, moet verder op zoek in een duister labyrinth.

Mandroid

In deze opvolger van het succesvolle Cyborg wordt je op een missie gestuurd om gestolen 'blueprints' terug te krijgen van de zeer geheime bouwwerken, functies en verdedigingssystemen. Dit document kan in handen van de verkeerde persoon een gevaarlijke uitwerking hebben.

Je gaat op zoek op diverse plaatsen, waar je verschrikkelijke tegenstanders ontmoet. Bandieten, dronkaards en gevallen vrouwen treden je tegemoet. Probeer ze om te kopen, maar pas op dat je ze niet tegen je in het harnas jaagt.

Naarmate het langer duurt om de blauwdrukken te achterhalen, produceren de boze krachten die ze ontvreemd hebben steeds meer Mandroids. Uiteindelijk moet je deze machines opsporen en buiten gevecht stellen. Dan moet je ook nog de kluis vinden waren de blauwdrukken bewaard worden.

Wraak

Dit spel mag misschien 'Vengeance' heten, 'Kamikaze' zou toepasselijker zijn geweest. Er wordt in dit spel van je verwacht dat je geheel alleen vijandelijke ruimteschepen aanvalt. Alleen een gek zou zo'n opdracht aannemen, want het enige resultaat kan een onmiddellijke dood zijn. Toch moet de buitenaardse terreur een halt toegevoerd worden, en iemand moet de verschrikkelijke vijand tegemoet treden op diens eigen terrein en volgens diens spelregels.

Wanneer je het buitenaardse ruimtevaartuig binnen komt, wacht je een afschrikwekkend doolhof van gangen en kamers. Als je alle benodigde componenten hebt verzameld kun je misschien je doel, wraak te nemen op de verschrikkelijke terreur, bereiken.

Timetrax

Om het heelal te redden is het in Timetrax noodzakelijk de Time Lords tevreden te stellen door ze acht dingen te overhandigen. Deze zaken moeten gevonden worden in het tijd/ruimte continuüm, dat je kunt bereizen door gebruik te maken van de mysterieuze 'Portals'. Deze tijdsuizen gaan op bepaalde tijden open, en als je die tijden eenmaal weet kunt je door de tijd heen en weer reizen. Daarbij kun je wel gehinderd worden door diverse rondtrekkende huurlingen die het slagen van je missie willen verhinderen en je zullen doden als ze de kans krijgen. Het spel bevat 21 verschillende grafische schermen, drie voor iedere tijd-zone. Als je het goed doet kun je van 'pudding' via 'reiziger' tot 'halfgod' promoveren.

Op jacht naar de schat

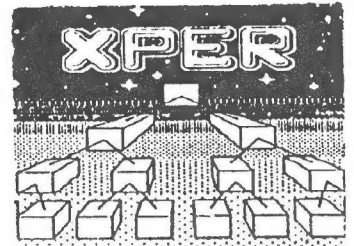
Wel wat intelligenter dan de verschillende vecht- ruimte en gedrochtenspelletjes is de 'Treasure Hunt' volgens het overbekende TV-principe. Gaat op TV een presentator fysiek op jacht naar de schat, terwijl de jagers in de studio moeten blijven, in dit spelletje dat op de Engelse 'jacht naar de schat' is gebaseerd, doe je het alle-

maal zelf. Op basis van de aanwijzingen moet je de piloot zeggen waar naar toe te vliegen en waar te landen om uiteindelijk de schat te vinden. Op het scherm kun je de ontwikkelingen bijhouden op een kaart waarop de afgelegde weg is te volgen, en een status-scherm geeft aan hoeveel tijd is verstreken en of je helicopter nog benzine heeft. Er zijn verschillende 'jachtpartijen' op de cassette bijgeleverd, waaronder een spel in Londen.

XPER 64

Met enige durf noemt men dit database pakket voor de C-64 nu al een expert systeem. Veel meer dan een vrij goed uitgedachte database met misschien wat Artificial Intelligence trekjes is het echter niet. Toch is dit pakket wel aan te raden, want voor de 64 zijn niet zo heel veel programma's op de markt, die een beetje tegen de professionele ontwikkelingen aanhangen. Het is een soort applicatie-generator ofwel een aanpak, waarbij op basis van vrij simpele gegevens en commando's een database opzet (structuur) wordt aangemaakt, waarmee men dan verder kan werken. Het vaak bewerkelijke opzetten van de file en record indeling met alle "fields" wordt

de gebruiker deels uit handen genomen, hij kan zich meer bezighouden met het vullen van de database, en het opvragen en wijzigen van de data. Dat manipuleren met gegevens, zoeken, sorteren en selecteren is een van de sterke punten van XPER-64. Dat heet dan heel fraai de rapporteringsfase in de database taal. Maar de er-



varen database-aanhanger weet, dat juist in het maken van die ogenschijnlijk zo simpele lijsten en uitdraaien, maar ook van stickers etc, heel wat werk gaat zitten. De prijs is niet echt aan de lage kant, Datahome verkoopt XPER-64 van Abacus (gelieerd aan Data Becker) voor f 299,50, maar het is zeker een interessante ontwikkeling op de 64, die bijvoorbeeld in het onderwijs de leerlingen snel vertrouwd kan maken met het nut en de beperkingen van een database. 020-837367.

The Living Daylights

Een full screen oog, een vervaarlijk heer in jacquet met Walther PK-pistool, een schot, bloed trekt over het witte doek en het James Bond-thema schalt de zaal in. James Bond is weer terug in bioscoop en als videogame. Op het witte doek schitteren de vierde "Bond" Timothy Dalton, Jeroen Krabbé als schurk, Bond-girl Maryam d'Abo en ook de zwaarbewapende Ashton Martin 160 Vantage is weer terug. Wie het allemaal thuis nog eens wil naspelen kan bij het Bond-huismerk Domark terecht. Dit Britse softwarehuis bracht haar traditie getrouw ook de nieuwste Bond-film, **The Living Daylights** als videogame uit. In het verleden waren wij niet onverdeeld gelukkig met de Bond- games van Domark. De graphics en spelactie vielen nogal eens tegen en een aantal versies bleken, tengevolge van de grote haast bij de release, verre van luisvrij. De laatste tijd heeft Domark haar leven echter gebeterd en het spelontwerp wordt tegenwoordig in handen van een vast machine-team gegeven. Zo is er een 68000 (Amiga en Atari ST)-, Commodore- en MSX-team. Dat geeft de koper meer zekerheid dat The Living Daylights uit zijn/haar computer haalt wat er in zit. Blijft over het spelgegevens. Een James Bond-film op het witte doek valt moeilijk na te vertellen. De uitgekende mixage van humoreus high tech-geweld, slimme super-

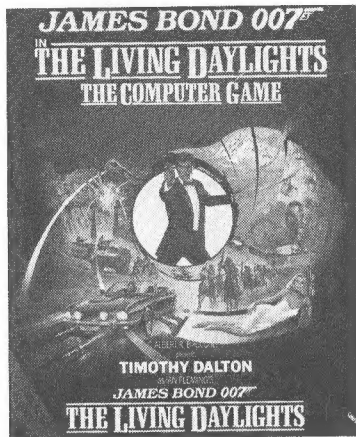
007

schurken, verleidelijk vrouwelijk schoon, de Bond-charme en ondersteunende muziek is al jaren een succesformule. De toeschouwer wordt als het ware in de werveling der gebeurtenissen meegesleurd. Deze succesformule in een videogame onderbrengen is een ondoenlijke zaak. De makers moeten zich noodgedwongen beperken tot een arcade-game met bijpassende muziek.

Het game The Living Daylights biedt de speler acht op de film gebaseerde spelniveaus in oplopende moeilijkheid. Ook het verhaal bleef identiek: De overloper Koskov moet uit Rusland ontsnappen voor de KGB-killers hem naar de andere wereld helpen. Helaas blijkt Koskov meer op het eigen voordeel dan dat van het vrije Westen uit. De slimme schurk heeft profijtelijke betrekkingen met de internationale wapenhandelaar Brad Whittaker. Voor het vrouwelijk schoon zorgt de fraai gevormde celliste Kara. Het sadistische geweld schuilt hem in de met koptelefoonkabels wurgende killer Necros. Terug naar de verschillende spelniveaus: * Niveau 1 is de introductie op **Gibraltar**. Een kleine test van de defensie op deze apenrots door commander Bond leert de speler de benodigde spelbehendigheid aan. De tegenstanders zouden slechts bevriende SAS-agenten zijn, maar er blijkt toch ook een verborgen killer tussen de goeden te zitten. Een beetje Bond weet zo'n snoodaard na-

tuurlijk wel te vatten. **M** heeft Bond in ieder geval voorzien van een ghetto-blasters (=radio) met ingebouwde raketten, pistool, Bazooka, granaten, nachtkijker, roestvrij-stalen hoed, mortieren, pennen met miniraketten en zelfs een kruisboog. De tegenstanders slaan terug met handvuurwapens, helicopters en exploderende melkflessen.

* Op niveau 2 wordt contact met Koskov gelegd. Helaas blijkt niet iedereen van muziek te houden en wemelt het dak van de **Lenin Music Conservatory** van de sluipschutters. Verdedig Koskov met de joystick tegen zijn belagers zonder andere muziekliefhebbers te raken.



* Niveau 3 behandelt de ontsnapping per **Trans-Siberische pijpleiding**. Hierbij moet Bond de arbeiders trachten te ontwijken.

* Op niveau 4, **The Mansion House** verschijnt Necros in Great Britain om Koskov te elimineren. Pas op voor de witte motor, want de melkman gooit met bommen.

* Niveau 5 gaat weer over Necros. Dit maal is de **Fairground** de plaats van handeling. Probeer contact met een collega te leggen en stel Necros buiten gevecht.

* In niveau 6 toont Koskov zijn ware gezicht. Op de daken van **Tangiers** ontbrand een verwoed gevecht tussen Bond, op zoek naar de ontsnapte Koskov, en de huurmoordenaars van de wapenhandelaar.

* Op niveau 7 wordt de zaak echt spannend. Bond zit in danig in het nauw in de Afghaanse woestijn. De plaats van handeling is een Russisch **militair complex** en de soldaten brengen meteen hun hele arsenaal in stelling.

* De afrekening in **Whittakers's huis** volgt op niveau 8. In een complete veldslag moet Bond "the master mind" ten val brengen.

The Living Daylights vinden wij als arcade spel redelijk tot geslaagd. De graphics zijn goed, de actie op de hogere niveaus boeit, spelen is voldoende moeilijk maar niet onmogelijk en de muziek mag gehoord worden. Helaas kan het game zoals gezegd de sfeer van de film niet evenaren. Verder is enig strategisch inzicht vereist voor de juiste wapenkeuze. Vergissingen in deze breken Bond zuur op hoewel tijdens het spel van wapen gewisseld kan worden. Enkele wapens zijn slechts één keer bruikbaar. Het statuspaneel informeert u over de beschikbare wapens, kracht en de nog overgebleven Bond-levens. Een joystick is noodzakelijk.

The Tube

De liefhebbers van Science Fiction-games kunnen weer eens ouderwets lekker smullen van Quicksilver's **The Tube**. De spelheld heeft de pech om met zijn/haar schip de Tracker II in een meteorenstorm terecht te komen. Daardoor ontstaat nogal wat ernstige schade die snel gerepareerd dient te worden. Om de ramp compleet te maken wordt de Tracker er en passant in een zwart gat gezogen en wachten aan het andere eind een club vijandige Aliens.

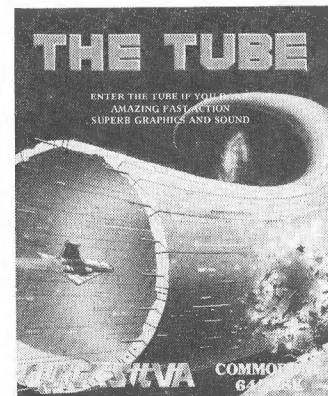
Klinkt bekend in de oren? Toch blijkt er een verassend nieuw spelidee in de The Tube te zitten. Achter het zwarte gat zit The Tube. Een krankzinnig tunnelstelsel opgezet door vreemde buitenaardse geesten met het doel om "oud ijzer" (lees onfortuinlijke ruimteschepen) te verzamelen. Een kunstmatig krachtveld begint de Tracker II al direct naar binnen te trekken en in de diepte van het Tube-complex dreigt een totale ontmanteling.

Gelukkig is het nieuwe slachtoffer, u dus!, wat scherper van geest dan de onfortuinlijke voorgangers. En met een goede strategie bij het aftappen van energie uit andere vaartuigen komt de speler een heel eind. Zet de defensieve krachtvelden alleen aan als het nodig is, voorkom botsingen met andere naar energie hunkerende objecten en plunder zelf zo veel mogelijk energiebronnen, dat zijn de sleutels tot overleven.

The Tube bestaat uit 8 segmenten met elk drie zones:

* De **Transfer Zone** is ontworpen om ruimteschepen te vangen, en vervolgens naar het centrum van The Tube te transporteren. Deze zone zit vol met parasitaire Aliens en rondzwervend schroot. Allen vreten uw energie zienderogen weg. Een slim bommetje of een laserbeschieting op zijn tijd schakelt tegenstanders snel en doeltreffend uit, maar kost zelf ook weer energie.

* De volgende zone is de **DMT**. DMT staat inderdaad voor een soort bestrijdingsmiddel. Alleen bestrijdt de Defence Mechanisme Tunnel geen ongedierte, maar loert de DMT op bewapende schepen die de Tube in gevaar kunnen brengen. Wordt een gevaarlijk schip gedetecteerd dan gaat de DMT onmiddellijk tot een zware aanval over. Schepen in slechte doen worden door dit Tube-deel verder verzwakt.



* Zij die de DMT overleven komen in de **Capture Zone**. Hier is het Mekka van de Alien ruimteslopers. Hier liggen de slachtoffers op hun vreemde meesters te wachten. U natuurlijk niet, want al die schepen kunnen nog bruikbare energiekristallen aan boord hebben. Voldoende gesterkt met energie kan de speler de volgende sector proberen. The Tube munt op de C-64 en C-128 uit door werkelijk sublieme graphics en de goed gekozen muziek van David Whittaker. De actie is glad en enerverend. Het game kan zowel met het toetsenbord als joystick gespeeld worden.

Modern Management

door: Bert tiér.



De beginnende programmeur kan met hulp van deze serie artikelen stap voor stap leren, hoe leuk het is, om zelf meer te doen met de Commodore, dan alleen het spelen van voorgekookte spelletjes en het gebruiken van andere gekochte software-applicaties. Stap voor stap worden in deze cursus door Jan Bodzinga de elementaire feiten van Basic behandeld. De stof is daarbij bewust erg eenvoudig en duidelijk gehouden, zodat ook de volslagen 'leek' op programmeer-gebied erdoor kan uitgroeien tot een redelijk programmeur. Maar ook de meer gevorderde 'Basiccer' zal ongetwijfeld nog wel dingen in deze cursus tegenkomen, waarmee z'n kennis kan worden aangevuld.



Basis Basic

Deel 17

De diskdrive

In de vorige aflevering zijn we tot in detail bezig geweest met het gebruik van de cassetterecorder. Om ook voor de Commodore diskdrive dezelfde informatie te kunnen geven, zullen we in dit artikel voornamelijk bezig zijn met het programmeren van dit externe device. Het maakt daarbij weinig verschil of je nu werkt met een C-16 en drive of bijvoorbeeld een C-128. Hoewel deze laatste beschikt over een serie geavanceerde Basic-opdrachten om met de drive en floppies te kunnen werken, zullen we voor de basiskennis uitsluitend gebruik maken van de elementaire Basic opdrachten, zoals die in 1976 door Commodore zijn ontwikkeld.

Tape of disk

Het grote verschil tussen de cassette-recorder en de disk-drive ligt in manier waarop beide de opgeslagen gegevens kunnen wegzetten en terug lezen. Laten we voorop stellen, dat **alles** wat met een cassettedeck kan worden uitgevoerd op (nagenoeg) dezelfde manier met de diskdrive kan worden gedaan. De enige uitzondering op deze stelling is wellicht de trucendoos die door sommige spelletjes-fabrikanten wordt open getrokken om met hulp van enkele interrupts de gebruiker tijdens het laden te vergastten op een leuk muziekje en wat

schermwerk. Bij gebruik van disks is het inlezen van software en data gelukkig lang niet zo tijdrovend, en deze trucs zijn daarbij ook minder nodig. Kortom, de echte verschillen, waarbij de cassette I/O tot meer in staat is, liggen op het niveau van expert-programma's. Wij zullen ons daar in dit geval niet mee bezig houden.

De diskdrive kan in ons geval dus alles uitvoeren, wat we met de casset-tetape ook kunnen doen. De tijd die dit bij de Commodore drives in beslag neemt is weliswaar niet erg kort, maar toch een heel stuk sneller dan we van de cassette gewend zijn. Maar dat is niet het enige voordeel van een disk-

drive ten opzichte van de tapeverwerking. Door de structuur van de diskette zijn we met hulp van de drive namelijk in staat om overal in het bestand te prikken en data te lezen. We noemen dat een relatieve benadering van bestanden. De cassettedeck kan alleen maar Byte voor Byte, sequentieel, de tape aftasten.

Bij de diskette kunnen we, net als bij een grammofoonplaat op de pick-up met de lees/schrijfkop overal op de disk gegevens lezen en schrijven. Daardoor is een disk ook sneller voor veel meer doeleinde aan te wenden. Hoewel tegenwoordig ook de disks niet zo duur meer zijn als een paar

jaar terug, toen er soms meer dan dertig gulden voor een floppy moest worden betaald, ligt de prijs van de diskdrive nog wel veel hoger dan van een eenvoudige cassette recorder. Maar de capaciteiten van de drive maken dit verschil zeker goed.

Bestand

Het fenomeen 'bestand' zal voor veel computergebruikers een begrip zijn, voor anderen is het nogal moeilijk een exacte voorstelling te maken over de juiste eigenschappen van een bestand. In principe is iedere verzameling gegevens (data) die op een opslagmedium is weggezet een bestand. Maar voor normaal gebruik is het natuurlijk beter, dat een bestand ook over een bepaalde structuur kan beschikken. Door deze structuur kunnen zowel de gebruiker als de drive sneller en efficiënter met een bestand omgaan. De naam 'file' wordt vaak gebruikt om een bestand aan te duiden en is dan ook precies hetzelfde. Om een filestructuur te geven, zullen we moeten weten wat dat inhoudt. Een verzameling data, samengebracht op tape of disk, is sowieso een bestand, maar ongestructureerd hebben we er niet zoveel aan.

Alleen tekstbestanden, uit een tekstverwerker, kunnen zonder structuur worden bewaard. De meeste andere bestanden zullen op de één of andere manier tot een logisch geheel moeten worden opgebouwd, om er later nog wat aan te hebben.

De eenvoudigste manier om dit te kunnen voorstellen is het oude voorbeeld van een kaartenbak. Het bestand op disk of tape kan daar het gemakkelijkst mee worden vergeleken. Alleen de namen van kaartenbak en bestand verschillen.

Laten we de kaartenbak maar eens beter bekijken, aan de hand van een adressenbestand. Voor ieder adres hebben we een kaart in de bak. Op elke kaart vinden we diverse vakken, waarin per vak een bepaald gegeven moet worden ingevuld. Zowel de plaats op de kaart als het kenmerk van het gegeven is daarbij vastgelegd. Dus voor een adressenbestand bijvoorbeeld linksboven op de kaart het **nummer**, daaronder de **naam**, vervolgens **straat**, **postcode** en **woonplaats**, eventueel nog aangevuld met **telefoonnummer**, **geboortedatum** en andere gegevens.

Eén dergelijke kaart bevat daarmee de volledige data van één adres. De volgende kaart ziet er natuurlijk net zo uit, maar bij de verschillende kenmerken vinden we op kaart twee uiteraard andere gegevens. Zo kunnen we in de

kaartenbak doorgaan tot er geen ruimte meer is, om de platte kaarten op te bergen.

Daarmee hebben we in feite een hardware-bestand gemaakt op de ouderwetse manier. Je kunt je voorstellen, dat deze wijze van het opslaan van gegevens nogal tijdrovend is. Zeker als je bedenkt hoe het in z'n werk zal gaan om de gegevens op alfabet en op postcode te moeten sorteren. Vandaar ook, dat de computer ons een heel stuk op weg kan helpen om dit procedé te versnellen.

In ieder geval kun je zien aan dit kaartenbakvoorbeeld, dat er een bepaalde structuur is aangebracht in de wijze waarop de gegevens op de kaart zijn geschreven. Het is niet chaotisch gebeurd met een willekeurig volgorde van de diverse kenmerken op de kaart, maar voor ieder gegeven is een apart vakje aanwezig, dat steeds op dezelfde plek op de kaart te vinden is. Het ging ons in dit geval om de vergelijking van de ouderwetse kaartenbak met een - modern - computerbestand. Het zal daarbij voor iedereen duidelijk zijn, dat het niet erg moeilijk is, om de computer (samen met diskdrive of cassette recorder) te gebruiken om de gegevens die op de verschillende kaarten worden bewaard te kunnen bewerken. Maar daarbij is de structuur van het bestand een belangrijk onderdeel.

Structuur

Het intypen van een hele serie gegevens, van bijvoorbeeld een adressenbestand, zal voor niet al te veel problemen zorgen. Wel is het daarbij nodig, zoals uit het voorbeeld met de kaartenbak blijkt, dat deze gegevens in een zekere structuur worden ingebracht en bewaard. Daardoor wordt het voor de computer erg gemakkelijk om de gegevens later op een correcte manier te kunnen lezen en bewerken. En dat is precies wat we willen. De structuur in het bestand is dus een erg noodzakelijk gegeven. We zullen ons daarom bezig houden met de vraag hoe een structuur is aan te brengen in een bepaald bestand.

Vooraf nog een beetje theorie, waarbij we de kaartenbak als leidraad zullen gebruiken. We zagen in de kaartenbak een hele serie kaarten, die stuk voor stuk de gegevens van één adres bevatten. In computertermen vertaald noemen we de kaartenbak een **databestand** en een dergelijke kaart met de volledige serie gegevens een **record**. Deze beide termen zijn standaard in de wereld van de informatica, en het is dan ook noodzakelijk om goed te weten wat ze betekenen. Er

komt nog een term bij, namelijk het **veld**. Als je het voorbeeld goed bekijkt, zal het niet moeilijk te raden zijn, waar een veld voor staat. Dat is de data die is ingevoerd bij de verschillende kenmerken, nummer, naam, straat, postcode, plaats etcetera, op de kaart in de bak.

Als we het geheel bekijken, dan zien we een structuur ontstaan zoals in schema 1 is uitgewerkt.

Op deze manier kunnen we een echte database bouwen, niet alleen in een kaartenbak, maar ook in de computer. Het maakt daarbij in ons geval nog weinig uit, of we de opslag op disk of tape sequentieel of relatief laten verlopen. Omdat we de database, die we in Basic nog steeds aan het programmeren zijn, zowel voor cassette- als diskgebruik geschikt willen maken, hebben we gekozen voor een sequentiele opslag. We zullen hier in het vervolg van dit artikel nog wel op terug komen.

Adressenbestand

Nu iets over de structuur die we in het

KAARTENBAK	DATABESTAND
kaart 1	record 1
kenmerken	velden
nummer	nummer
naam	naam
straat	straat
postcode	postcode
plaats	plaats
kaart 2	record 2
kenmerken	velden
nummer	nummer
naam	naam
straat	straat
postcode	postcode
plaats	plaats
kaart 3	record 3
kenmerken	velden
nummer	nummer
naam	naam
straat	straat
postcode	postcode
plaats	plaats
kaart 4	record 4
kaart 5	record 5
kaart 6	record 6
enz.	enz.

Schema 1: de structuur van een kaartenbak

programma 'DATABASE' hebben toegepast. Als je een oude Commodore-Info openslaat, kun je in de listing duidelijk de structuur van ons bestand teruglezen. Ook de diverse variabelen, die we voor de verschillende records en velden gebruiken vallen daaruit te halen. Voor de duidelijkheid zullen we een paar delen uit de listing lichten en nog eens bekijken:

```
10 REM Menu gestuurde
  DATABASE
20 REM voor adressenbestand
30 REM Commodore-Info 1987
40 REM 870601/ v.1.05
100 REM initialisatie
110 REM
120 TT=0 : REM totaal aantal
    records
130 DIM A$(300,5) : REM aantal
    records max.
```

In de kop van het programma zien we de tweedimensionale **array A\$()**. Deze wordt in het programma gebruikt, om de volledige gegevens van ons bestand in het computergeheugen te kunnen bewaren. Zoals in regel 130 te zien is, hebben we ruimte gedefinieerd voor maximaal 301 adressen, die ieder over 6 velden kunnen beschikken. Daarbij bewaart A\$(0,0) veld nummer 1 van record 1, terwijl bijvoorbeeld A\$(34,4) veld nummer 5 van record 35 aangeeft.

De verschillende velden hebben we voor de duidelijkheid van de listing in DATA-regels weggezet:

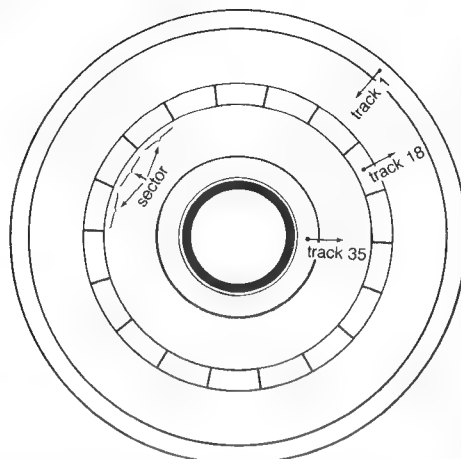
```
420 DATA "Naam"      "
421 DATA "Voorletters" "
422 DATA "Straat en nummer" "
423 DATA "Postcode"   "
424 DATA "Woonplaats" "
425 DATA "Telefoon"   "
```

We zien daarbij precies 6 velden, die altijd terug grijpen op de array A\$(), en dan ook nog op dezelfde plaats in de tweede dimensie van de array. Daarmee hebben we de structuur van ons bestand op een redelijke wijze voor elkaar.

Sequentieel bestand

Zoals we eerder hebben opgemerkt, is het 'sequentiële' bestand weliswaar correct wat de structuur betreft, maar we kunnen deze wijze van opslag niet gebruiken, om met de zogenoemde random-access een bepaald record terug te lezen. Om dit te kunnen begrijpen, zullen we onze toevlucht nemen tot een voorbeeld. De sequentiële wijze van opslag houdt in, dat alle gegevens, records en velden **achter elkaar** op disk of tape worden gezet.

TRACK NUMBER	SECTOR NUMBER	SECTOREN / TRACK
1 - 17	0 - 20	21
18 - 24	0 - 18	19
25 - 30	0 - 17	18
31 - 35	0 - 16	17



De scheiding tussen de verschillende velden en records kan daarbij worden aangegeven met een carriage return (CHR\$(13)) of bijvoorbeeld een komma. Dit scheidingsteken kan in veel gevallen door de programmeur worden gekozen, maar moet natuurlijk niet voorkomen in een veld zelf, anders gaat de zaak fout. Een eenvoudig bestand, met records die bestaan uit naam, straat, plaats kan er weggeschreven bijvoorbeeld zo uit zien:

```
Gerritsma,Dorpsstraat 45,
Groningen,Bruintjes,Bosweg 15,
's-Graveland,Bos,Klap 6,Twist,
Heertjegrootmans,
Asch van Wijkstraat 34,
Heerhugowaard.
```

We zien daarbij direct, dat de verschillende velden en dus ook de records niet allemaal even lang zijn. Duidelijker wordt dat nog, als we de records onder elkaar zetten:

```
#1 Gerritsma,Dorpsstraat 45,
  Groningen,
#2 Bruintjes,Bosweg 15,
  's-Graveland,
#3 Bos,Klap 6,Twist,
#4 Heertjegrootmans,Asch van
  Wijkstraat 34,
  Heerhugowaard.
```

Als we gaan tellen, blijkt, dat record 1 een lengte heeft van 32 karakters, record 2 bevat er 30, nummer 3 heeft slechts 14 karakters, terwijl record 4 maar liefst 51 letters bevat.

Als we dezelfde telling uitvoeren op de diverse velden, zien we dat daar nog meer verschillen optreden. De hier getoonde manier laat zien, hoe de verschillende records, gescheiden door

komma's, op tape of disk als sequentieel bestand (achter elkaar) kunnen worden weggeschreven. Daarbij begint record 1 op positie (offset) 0 in het bestand, record 2 begint op 0+32= pos. 32, record 3 op 0+30+32= 62, record 4 begint op pos 0+30+32+14= 76 terwijl record 5 een offset krijgt van 0+30+32+14+51=127. Je ziet direct, dat er in een sequentieel bestand geen pijl op valt te trekken waar het volgende record precies begint. Laat staan dat het voor de verschillende velden in een record het geval zou zijn.

Daardoor zullen we om record 4 te kunnen lezen het hele bestand vanaf record 1 moeten lezen en eventueel weggooien om nummer 4 in het geheugen te kunnen halen. Nu is dat bij een dergelijk klein bestand niet erg, want het gaat nogal snel, maar als we praten over 10.000 adressen wordt dat natuurlijk anders. Voor grote bestanden kunnen we dan ook echt niet meer volstaan met een sequentiële opslag zoals die hierboven is omschreven. Ook als we een wijziging in record 3 aan willen brengen, kunnen we niet volstaan met het tussenvoegen in een sequentieel bestand. Stel dat door de mutatie de lengte van record 3 wordt vergroot van 14 naar 26 letters. Dan zou-voorzover het technisch mogelijk is op tape- record 4 voor een gedeelte door de verlengde record 3 worden overschreven, en daardoor niet meer leesbaar zijn, afgezien van de overig problemen. Hieruit blijkt, dat er nogal wat problemen komen kijken bij deze vorm van data-opslag, ware het niet, dat deze manier veruit de eenvoudigste vorm is om te programmeren.

Databestand SEQ

Het werken met diskdrive en databestanden is niet veel anders, dan de werkwijze met Basic-programma's en disk. In de vorige les hebben we hier voor tape-gebruik het nodige over gezegd, voor de diskdrive ligt het niet veel anders. Volledigheidshalve zullen we ons geheugen over de bestandenprocedure nog wat oprispen.

We maken bij het wegschrijven van gegevens echter geen gebruik van de disk-SAVE en LOAD-opdrachten, omdat we met een andere manier van schrijven en lezen bezig zijn. De wijze waarop DATA en disk met elkaar communiceren kun je wat dit betreft het beste vergelijken met de manier waarop de Commodore met het videoscherm (en toetsenbord) omgaat. We weten allemaal, dat met hulp van de opdrachten PRINT, GET en INPUT van alles op het scherm kan worden

gezet en vanaf het toetsenbord kan worden ingelezen. Welnu, de manier waarop dit met externe apparaten als cassette recorder en diskdrive wordt verzorgd is in feite niet anders. Het enige verschil is het **device-nummer** dat voor ieder apparaat, dus ook voor scherm en keyboard, anders is. Omdat er in de meeste gevallen communicatie tussen computer en VDU (scherm/keyboard) plaatsvindt, heeft men dit device als **default** genomen, en alle print-opdrachten gaan daarom automatisch naar device nummer 3, wat het nummer is van het videoscherm. Voor alle andere apparaten moeten we in ons programma een apart device aanspreken, en een I/O kanaal openen. Dit klinkt een heel stuk ingewikkelder dan het in werkelijkheid is, maar je moet wel eerst weten hoe het allemaal precies in z'n werk gaat. Zeker omdat het aantal kanalen dat op een bepaald moment open kan zijn, niet bij één blijft. In totaal kunnen er 16 kanalen gelijktijdig geopend zijn, die alle 16 naar verschillende devices kunnen wijzen. Zo kennen we voor de printer (device 4) een kanaal, terwijl er ook meerdere kanalen open kunnen zijn voor cassette en diskdrive. Om dit allemaal uit elkaar te kunnen houden, geven we bij het openen ieder kanaal een apart nummer of adres. Met hulp van dit 'Logical Address' geven we later aan naar welk kanaal een bepaalde variabele moet worden geprint. Het werken met deze kanaal-adressen is een goede manier om de verschillende kanalen uit elkaar te kunnen houden. De algemene werkwijze van het versturen van berichten c.q. data naar een extern apparaat is als volgt:

OPEN file, met kanaalnummer, apparaatnummer en secundair adres.
PRINT kanaalnummer, variabelen
CLOSE kanaalnummer

Het secundaire adres dient bij diskcommunicatie, in tegenstelling tot dat van de cassette, in feite nergens toe. Wel wordt er door de Commodore onderscheid gemaakt tussen SA's die hoger zijn dan 127 en alles wat er onder valt, maar dat is te complex voor dit artikel. Alleen SA 15 maakt hierop een uitzondering. Bij het geven van een bepaald 'secundair adres' moet je er alleen om denken, dat er geen verschil zit tussen lezen en schrijven naar de disk.

Schrijven

Laten we maar eens een klein programma schrijven, waarmee we een

serie gegevens naar de cassette weg schrijven.

```

10 REM Maken data bestand op disk
20 REM Sequentieel bestand
30 REM Met numerieke variabele
90 :
100 REM Openen kanaal
110 OPEN 2,8,2,"TEST,SEQ,WRITE":REM Openen KANAAL=2, DEVICE=8, SA=2, TEST=naam
115 REM Let op toevoeging SEQ,WRITE
120 FOR I = 1 TO 256
130 PRINT I : REM Print op scherm
140 PRINT#2,I : Rem print naar kanaal 2
150 NEXT I
160 CLOSE 2 :REM Sluiten KANAAL 2
200 END

```

Dit programma schrijft zowel op het scherm als naar de disk 256 getallen weg. De naam van het DATA-bestand op de disk directory is in dit geval 'TEST'. In tegenstelling tot de tape-listing, zien we dat tussen de aanhangstekens de toevoeging: 'SEQ, WRITE' is mee gegeven, waardoor we aan de computer duidelijk maken, dat het een sequentieel bestand betreft, wat we willen aanmaken en vol schrijven.

Na de I-lus wordt in regel 160 het kanaal naar device 8, de diskdrive, nummer 2 weer afgesloten. Naast de OPEN opdracht in regel 110 vinden we in regel 140 : PRINT#2,I. Deze regel, die volledig moet worden ingetypt, dus geen vraagteken (?) gebruiken voor het woordt PRINT !!!, geeft aan dat de variabele I moet worden geprint naar kanaal 2, en niet naar het scherm. Na de CLOSE-opdracht in regel 160 wordt kanaal 2 weer afgesloten en staat bestand 'TEST' op de disk. Dan verschijnt ook READY op het scherm, om aan te geven, dat het programma is afgerond. We kunnen dit nog even controleren, door de opdracht :

LOAD \$,8, waarna bij het commando LIST de inhoud van de diskette op het scherm komt, waar we file TEST met extensie SEQ zullen aantreffen.

Lezen

Nadat we het voorgaande met succes hebben uitgevoerd, kunnen we nu overgaan tot het lezen van de weggeschreven getallen in file TEST. Zorg dat de disk nog in de drive aanwezig is en gebruik het volgende programma om de weggeschreven gegevens weer te kunnen inlezen:

```

10 REM Lezen data bestand van disk
20 REM Sequentieel bestand
30 REM Gebruik numerieke var.
90 :
100 REM Openen KANAAL
110 OPEN 3,8,8,"TEST,SEQ,READ":REM Openen KANAAL=3, DEVICE=8, SA=8, TEST=naam
115 REM Let op : SEQ,READ
120 INPUT#3,I :REM Lees van KANAAL 3
150 PRINT I : REM Zet gelezen getal op scherm
160 IF ST = 0 THEN 120
170 CLOSE 3 :REM Sluiten KANAAL 3
200 END

```

Dit programma ziet er een beetje anders uit, dan het voorgaande. Het OPEN zetten van het kanaal naar de disk verschilt in zoverre, dat het SA (secundaire adres) gewijzigd is in 8, waarmee we willen aantonen, dat de keuze van een SA niet belangrijk is. Het feit, dat we willen lezen in plaat van schrijven, vinden we terug in de tekst, achter de filenaam in het openingsstatement. Als regel 110 gelijk zou zijn aan dezelfde regel in het schrijfprogramma, dan zou er een diskfout zijn opgetreden, omdat file TEST reeds op de disk aanwezig is. In ieder geval wordt er dan niets gelezen.

In regel 120 treffen we nu een INPUT-opdracht aan, net als bij PRINT gevolgd door #3. De Basic opdracht is geen combinatie van INPUT en #3, maar luidt echt INPUT#, met daarachter het betreffende filekanaal. De gelezen waarde, in dit geval numeriek, wordt weggezet in de variabele I, waarna deze op het scherm geprint wordt. INPUT houdt automatisch op te lezen bij het vinden van een 'carriage return' (CHR\$(13)). Net als bij een normale PRINT opdracht hebben we een CHR\$(13) weggeschreven bij het vorige programma, omdat we steeds een PRINT#3 hebben uitgevoerd zonder als achtervoegsel een komma of puntkomma te hebben gebruikt. Automatisch wordt dan ook naar disk en tape een carriage return geschreven. Vandaar dat de INPUT nu keurig steeds één getal naar binnen leest en op het scherm zet.

Status

Om te kunnen bekijken waar het einde van het bestand TEST zich bevindt, hebben we in ons geval verschillende mogelijkheden. We weten bijvoorbeeld, dat er precies 256 getallen in

het bestand staan. Daarom zouden we hier voor het inlezen ook een lus kunnen gebruiken. Maar het zal zeker niet altijd het geval zijn, dat we vooraf het aantal gegevens weten, dat in een database zit. Bovendien hadden we bijvoorbeeld als eindvlag een negatief getal kunnen wegschrijven en tijdens het inlezen iedere keer het gelezen getal hebben kunnen testen op het al of niet kleiner dan nul zijn. Maar daarmee beperken we ons tot het werken met uitsluitend positieve getallen, ook geen oplossing dus. Nu heeft de disk gelukkig zelf een oplossing. En dat is de status van de file/tapeunit. Deze STatus is één van de vaste, ingebouwde variabelen in de Commodore. Het is een variabele die door de programmeur alleen kan worden gelezen.

In ons lees-programma testen we slechts of ST nog altijd 0 is. Zolang dit het geval is, is alles in orde en kunnen we terug naar regel 120 om het volgende getal te lezen.

Bereikt de leeskop op de disk het einde van het bestand, waar door de Commodore automatisch een **end of file** geschreven is tijdens het wegzetten van de data, dan wordt de variabele ST automatisch op een getal anders dan nul gezet, waardoor we in regel 170 het file afsluiten door CLOSE en ook de drive stopt. Zoals je op het scherm kunt zien, zijn alle getallen keurig ingelezen van het disk-bestand 'TEST'.

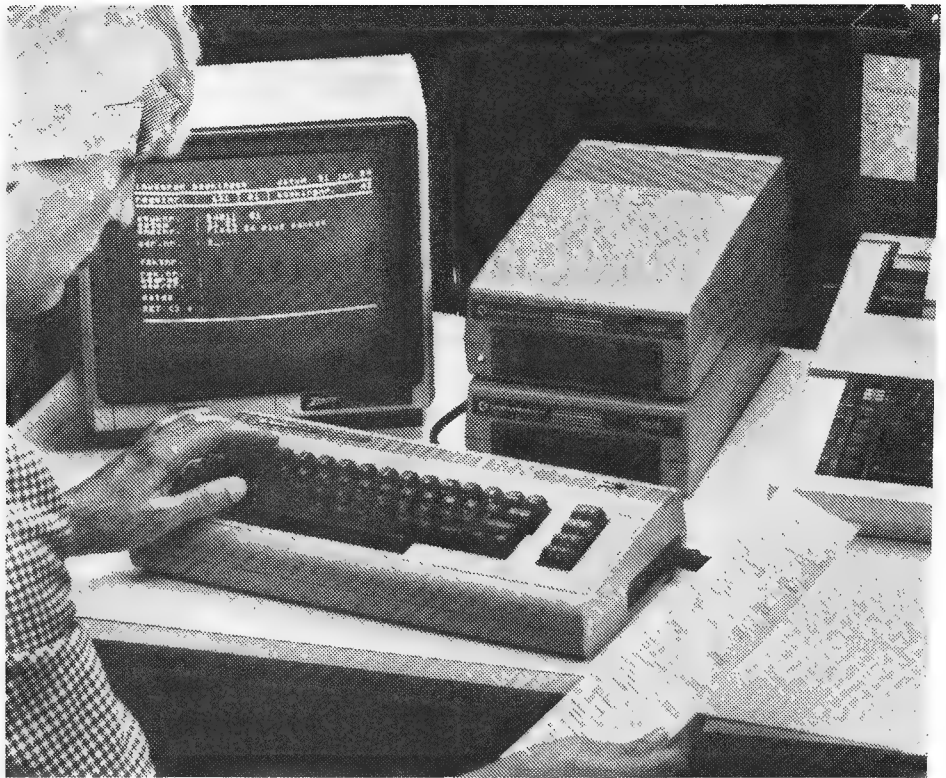
Strings

We zullen hetzelfde bestand nog eens inlezen, nu niet in een numerieke variabele, maar met hulp van strings. Daartoe kun je het vorige programma met wat kleine wijzigingen gebruiken:

```

10 REM Lezen data bestand van
   disk
20 REM Sequentieel bestand
30 REM Gebruik strings
90 :
100 REM Openen KANAAL
110 OPEN
    1,8,2,"TEST,SEQ,READ"
    :REM Openen KANAAL=1,
    DEVICE=8, SA=2,
    TEST=naam
120 INPUT#1,A$:REM Lees van
    KANAAL 1
150 PRINT A$ : REM Zet gelezen
    getal op scherm
160 IF ST = 0 THEN 120
170 CLOSE 1 :REM Sluiten
    KANAAL 1
200 END
  
```

Zoals je ziet, zijn de numerieke variabelen vervangen door de stringvariabelen.



bele A\$. Ook op het scherm is dit te zien, door de andere layout van de getallen, die nu immers als strings worden beschouwd, waardoor er geen ruimte wordt vrijgehouden voor een plus- of min-teken etcetera. Verder is er nog weinig veranderd. Maar we zullen nu een ander bestand naar de tape zetten, waarbij er ook alfanumerieken worden weggeschreven naar bestand TEST. Daarvoor gebruiken we de volgende variant op onze listing:

```

10 REM Maken data bestand op
   disk
20 REM Sequentieel bestand
30 REM Strings en numerieken
90 :
100 REM Openen KANAAL
110 OPEN
    2,8,2,"TEST1,SEQ,READ"
    :REM Openen KANAAL=2,
    DEVICE=8, SA=2,
    TEST1=naam
120 FOR I = 1 TO 256
125 A$ = CHR$(I)
130 PRINT I, A$ : REM Print op
    scherm
140 PRINT#2,I, A$:REM Print naar
    KANAAL 2
150 NEXT I
160 CLOSE 2 :REM Sluiten
    KANAAL 1
200 END
  
```

Dit programma schrijft 2 keer zoveel gegevens naar de tape. Niet alleen de lus-teller, maar ook de ASCII-waarde

als karakter wordt naar de disk geschreven. Daarmee zijn we zeker van een databestand, dat een mixture is van numerieke en string-variabelen. Om dit bestand terug te lezen moeten we het laatste LEES-programma gebruiken, waarbij de naam TEST moet worden veranderd in TEST1, de listing die werkt met strings dus, want als we de numerieke versie hierop loslaten, dan krijgen we erg snel de foutmelding 'DATA MISMATCH ERROR', waarmee wordt aangeduid, dat er numerieke gegevens werden verwacht en een alfa-karakter werd gelezen, dat niet kan worden verwerkt.

Protocol

Bij het werken met databestanden is het in alle gevallen beter om uitsluitend met stringvariabelen te werken. Hoewel er ruimtebesparend kan worden gewerkt met het toepassen van numerieke variabelen is de kans op lees- en schrijffouten hoger, terwijl ook de gebruiksmogelijkheden van numerieke data uiteraard geringer zijn. Het is daarom in de regel goed alleen met stringvariabelen te werken in combinatie met data-bestanden.

Als laatste voorbeeld geven we nu nog een listing waarbij het GET# commando is toegepast:

```

10 REM Lezen data bestand van
   tape
20 REM Sequentieel bestand
  
```

```

30 REM Stringvariabelen
40 REM Met hulp van GET#
   opdracht
90 :
100 REM Openen KANAAL
110 OPEN 2,8,2,"TEST1" :REM
   Openen
120 GET#2,A$ :REM Lees 1
   karakter van KANAAL 2
150 PRINT A$; : REM Zet gelezen
   getal op scherm
160 IF ST = 0 THEN 120
170 CLOSE 2 :REM Sluiten
   KANAAL 2
200 END

```

Ook deze listing wijkt weinig af van de vorige. Let alleen op de puntkomma achter het PRINT-commando in regel 150. Daardoor worden alle karakters achter elkaar op het scherm gezet. De 'RETURN', die ook van de disk wordt gelezen, zorgt daarbij op de juiste plaats voor een nieuwe regel. GET# wordt vaak gebruikt als er geen enkel gegeven bestaat over de manier waarop de data in het bestand is opgebouwd. We hebben hiermee de manier van werken met sequentiële bestanden voldoende bekeken, denk ik.

Relatieve bestanden

Om dezelfde trucs uit te halen met relatieve bestanden komt er wat meer kijken. Maar eerst zullen we moeten weten, wat een dergelijk bestand precies inhoudt. Misschien helpt het, als je weet dat relatieve bestanden ook vaak worden aangeduid als 'random access' files, wat zoveel betekent als 'willekeurig te benaderen'.

Eerder in deze les hebben we een serie records van verschillende lengtes ingetypt, waarbij we zagen dat de offset van elk record anders was. Willen we daar toch een relatief bestand van maken, dan moeten we in feite alleen zorgen, dat ieder record precies hetzelfde aantal Bytes bevat. Als je dit goed bekijkt, betekent het dus in feite, dat een relatief record, waarmee we veel sneller en efficiënter kunnen werken, ook niet meer is dan een echt 'sequentiële' bestand. Alleen de offset van ieder record moet kunnen worden berekend, zodat alle records bij relatieve bestanden even lang moeten zijn. We kunnen dit het simpelst oplossen, door bij ieder record de velden op te vullen met spaties, tot ze de juiste lengte hebben gekregen.

Voor dat we zoiets kunnen gaan doen, zullen we onszelf een beperking moeten opleggen, waar het de maximale lengte per veld of per record betreft. Daar hebben we bij sequentiële bestanden weinig mee te maken, omdat het niet uitmaakt hoe lang een be-

paald record wordt. Ook de ruimte die een relatief bestand op disk inneemt is minder economisch, omdat er nogal wat nutteloze spaties in zullen voorkomen om de records op juiste lengte te brengen.

Als we het helemaal echt zouden willen doen, moeten we uiteraard per veld een maximum lengte zetten, maar in principe komt het voor de huis-, tuin- en keuken toepassingen in de Commodore op hetzelfde neer, als we per record een evenlang record aanhouden, waarin de lengte van de velden afzonderlijk er niet toe doet. De velden kunnen blijvend gescheiden worden door komma's.

Hoe de records er voor een relatief bestand uit komen te zien zie je hieronder. In plaats van spaties hebben we een liggend streepje gebruikt om ze aan te geven. Je kunt daardoor goed zien, dat ze allemaal even lang geworden zijn.

REL. bestand

In tegenstelling tot de cassette tape, kunnen we op de disk wel midden in een bestand 'prikken' met de lees-/schrijfkop, waardoor het mogelijk wordt om midden in een relatief (random access) bestand te lezen en te schrijven, zonder daarbij het hele bestand te hoeven dupliceren, zoals dat bij sequentiële wijzigingen nodig is. Commodore heeft in eerste instantie helaas niet al te veel met dit soort mogelijkheden gerekend, waardoor het bij de eerste versies Basic en BIOS van de diskdrive niet of nauwelijks mogelijk is om met een dergelijk bestandstype te werken. Vanaf de business-machines 8032 en ook de C-128 kunnen we met Basic 4.0 wel aan het werk met geavanceerde opdrachten als DOPEN, RECORD# etc, waarmee we het werken met relatieve bestanden tot een peuleschil maken. Als voorbeeld een korte listing zonder uitleg, waarmee van bestand 'RELTEST' in één keer 100 records worden weggeschreven op de disk:

```

10 REM voorbeeld Basic 4.0
   REL.BESTAND schrijven
90 :
100 DOPEN#2,"RELTEST",D0,L20
   :REM Lengte record=20
110 FOR I= 1 TO 100
120 RECORD#2,(I) :REM Zoek
   record

```

```

130 $$= "RECORD NR :"+STR$(I)
140 $$+LEFT$("*****",20-LEN($$)
   *****",20-LEN($$)
150 PRINT#2,$$
160 NEXT I
170 DCLOSE
180 END

```

Regel 130 en 140 zorgen samen dat er vulling komt voor de records van 20 posities lang. In plaats van spaties hebben we '***' gebruikt voor het opvullen.

Het teruglezen, voor de verandering vanaf record 100, gaat nagenoeg identiek:

```

10 REM voorbeeld Basic 4.0
   REL.BESTAND lezen
90 :
100 DOPEN#2,"RELTEST",D0
   :REM Lengte is bekend
110 FOR I= 100 TO 1 STEP -1
120 RECORD#2,(I) :REM Zoek
   record
130 INPUT#2,$$
140 PRINT $$
160 NEXT I
170 DCLOSE
180 END

```

Voor het feit dat de records vanaf nummer 100 kunnen worden teruggelezen geeft hier een duidelijke illustratie van de capaciteiten die schuilen in relatieve bestanden. Maar ook Basic 4.0 en hoger hebben nogal wat gemak opgeleverd bij het programmeren van deze listings, omdat we gebruik gemaakt hebben van speciale opdrachten, die we op geen enkele kale C-64 tegenkomen. Willen we op dat type apparaat toch wat gaan doen met REL. bestanden, dan moeten we helaas uit een ander, complexer vaatje gaan programmeren.

C-64 Basic

De Basic versie die ten tijde van de ontwikkeling van de C-64 gangbaar was voor Commodore is niet in staat zonder meer met relatieve bestanden aan het werk te gaan.

Toch kunnen we bij de aansluiting van de juiste drive met kunst en vliegwerk aan de gang met relatieve bestanden. Als voorbeeld hebben we de listings van Basic 4.0 bewerkt voor alle ongelukkige bezitters van wat eenvoudiger apparaten:

#1	"Gerritsma, Dorpsstraat 45, Groningen-----"
#2	"Bruintjes, Bosweg 15, 's-Graveland-----"
#3	"Bos, Klap 6, Twist-----"
#4	"Heertjegrootmans, Asch van Wijckstraat 34, Heerhugowaard---"


```

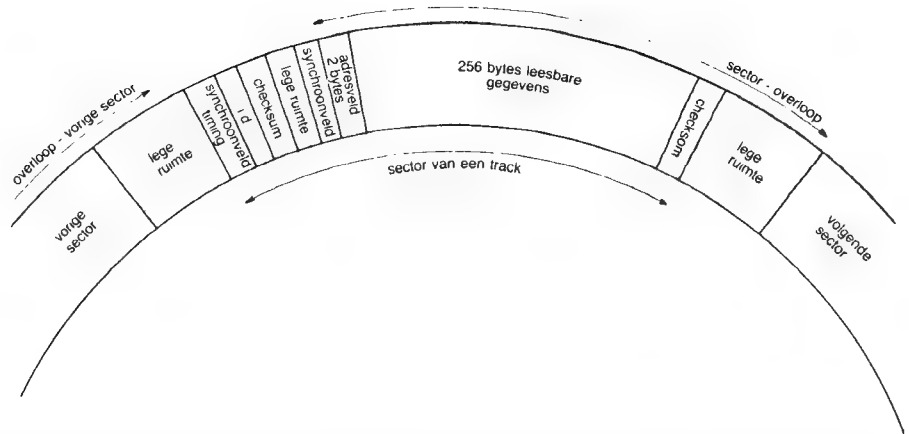
10 REM voorbeeld Basic 2.0
REL.BESTAND schrijven
90 :
100 OPEN 2,8,2,"RELTEST,L"+
CHR$(21):REM Lengte
record=20+1
105 OPEN 15,8,15...:REM Apart
KANAAL 15 !!
110 FOR I= 1 TO 100
120 PRINT#15,"P" + CHR$(2) +
CHR$(I) + CHR$(0) +
CHR$(1):REM Record
130 $$= "RECORD NR : " + STR$(I)
140 $$+LEFT$("*****",20-LEN($$)
*****",20-LEN($$)
150 PRINT#2,$$
160 NEXT
170 CLOSE 2
175 CLOSE 15
180 END

```

Hoewel veel ingewikkelder, en moeilijker te onthouden, kan met deze syntax ook voor de C-64 een fatsoenlijk relatief bestand op disk worden gezet.

Het lijkt me goed, om het OPEN statement even onder de loop te nemen. Zoals je ziet hebben we twee kanalen moeten openen om met relatieve bestanden te werken. Niet alleen kanaal 2, maar ook kanaal 15, wat vanwege z'n SA (15) een erg bijzondere functie heeft in de communicatie tussen computer(programma) en diskdrive. Het is een zogenoemd 'Fout-kanaal'. Er zou nog een heleboel over 'channel15' te vertellen zijn, maar daarvoor hebben we op dit moment geen mogelijkheden. Je moet dit SA voor het moment beschouwen als een soort politie-agent, waarbij je eigenlijk alleen moet zorgen dat SA 15 niet mag worden vervangen door een ander adres. Via dit kanaal worden de aanvullende opdrachten voor de relatieve bestanden doorgegeven aan de diskdrive. Het is zelfs zo, dat ook de 'geavanceerde' Basic 4.0 opdrachten, als DOPEN, RECORD, DIRECTORY etcetera allemaal met hulp van SA 15 in het Commodore-interpreter gedeelte worden vertaald naar dit type opdrachten, zoals we die in de voorgaande listing aantreffen.

Kanaal 15 wordt met OPEN 15,8,15 in regel 105 open gezet. Daarvoor is in regel 100 het 'echte' relatieve bestand geopend door de opdrachtreeks: OPEN 2,8,2,"RELTEST,L"+CHR\$(21). Hiermee wordt aangegeven, dat bestand RELTEST op drive 0 moet worden geopend, met een recordlengte "L"+CHR\$(21). Dit komt neer op recordlengte 20 + 1 voor



de carriage return = 21. We kunnen naast de RETURN dus 20 karakters kwijt in het record.

Via regel 120 zoeken we in het bestand het goede record, door de opdracht : PRINT#15,"P" + CHR\$(2) + CHR\$(I) + CHR\$(0) + CHR\$(1). De enige variabele die we tegenkomen in deze serie karakterstrings is I, waarmee het recordnummer in het bestand wordt aangegeven. Voor de rest wijzigt er niets in deze regel, en ze hoeft daarom dus ook maar één keer uit het hoofd te worden geleerd.

Lezen C-64

Uiteraard wil je ook kunnen controleren, of de gegevens werkelijk (relatief) zijn weggezet op de disk. Met dat doel hebben we het schrijfprogramma omgewerkt, zodat het hele scala met de volgende listing wordt gecompleteerd:

```

10 REM voorbeeld Basic 2.0
REL.BESTAND lezen
90 :
100 OPEN 2,8,2,"RELTEST":REM
LENGTE bekend
105 OPEN 15,8,15...:REM APART
KANAAL 15 !!
110 INPUT "RECORDNUMMER
(1-100/ 0=stop)";I
115 IF I = 0 THEN 170 :REM END
120 GOSUB 200 :REM SPLITS I
130 PRINT#15,"P" + CHR$(2) +
CHR$(RL) + CHR$(RH) +
CHR$(0) + CHR$(1):REM
RECORD
140 INPUT#2,$$
150 PRINT $$
160 GOTO 110
170 CLOSE 2
175 CLOSE 15
180 END
190 :
200 REM SUBROUTINE
210 REM Uitrekenen recordbytes

```

```

220 REM RL en RH vanaf I
230 RH=I/256 :REM HI-BYTE I
240 RL=I AND 255 :REM
LO-BYTE I
250 RETURN

```

We hebben de listing ook aangepast door in plaats van de loop nu aan de gebruiker te vragen welk recordnummer er moet worden gelezen. Denk er om, dat in deze eenvoudige listings niets wordt gedaan aan het testen van de diskdrive. Geef je bijvoorbeeld als record '150' in, dan leest de drive op de disk in dit bestand niets goeds, omdat record 150 niet bestaat. Verder kun je deze met deze serie listings fatsoenlijk aan de slag om je eigen applicaties te verwerken.

In de volgende aflevering zullen we de informatie met betrekking tot tape en disk eindelijk verwerken in ons eigen database programma, zodat we kunnen beschikken over een mooi afgerond pakket. Je kunt natuurlijk met deze kennis zelf ook alvast aan de slag.

Sterkte.

Jan Bodzinga

**Abonnement op dit
blad ?**

Bel (gratis):

06-0224222

elke dag tot 20.30
(ook in het weekend)

Rob Goudriaan bespreekt weer enkele klassiekers onder de computerspelletjes voor de Commodore 64. Sommige van deze toppers zijn nog gewoon in de winkel te koop, naar anderen zult u moeten zoeken.

OUD VAN GOUDriaan

Castells of Doctor Creep.

In dit grafische adventure-spel in de vrij klassieke kasteel-opzet bent u een gevangene. De boosdoener is Doctor Creep, uw aartsvijand. Het grote voordeel is dat u zelf mag kiezen in welk van zijn kastelen u gevangen zit. In het kasteel angekommen wilt u natuurlijk zo gauw mogelijk proberen te ontsnappen. Daarvoor moeten vele hindernissen overwonnen worden.

Alle kastelen bestaan uit diverse kamers en iedere kamer heeft zijn eigen specifieke hindernissen. Een aantal deuren gaan niet zo maar open, er moet soms eerst naar een sleutel worden gezocht, de kleur van de sleutel stemt overeen met de kleur van de deur. Zoals een goed kasteelheer betaamt laat hij zijn gevangenen niet zo maar ontsnappen. Er wordt gebruik gemaakt van de meest moderne bewakingsapparatuur.

Telkens bij het verlaten van een kamer verschijnt er een plattegrond met daarop een knipperend pijltje. Hieraan is te zien waar u bent en wat de richting is die gevolgd moet worden. Eén van de moeilijkere hindernissen is de elektrisch geladen bol. Deze richt bliksemstralen naar beneden en een aanraking is absoluut dodelijk. Er zijn een aantal schakelpanelen waarmee deze bollen zijn te vernietigen dan wel zijn te verplaatsen naar een hoek waar je niet meer hoeft te zijn. Het kan zijn dat een kamer meerdere keren moet worden betreden voor een sleutel bemachtigd is, of een uitgang kan worden bereikt. Een teleporteermachine is in staat je te verplaatsen naar een gekleurde bol, en kan dus een prettig hulpmiddel zijn. Kamers met valluiken zijn erg gevaarlijk, zij zijn open of dicht te schuiven met een schakelaar die zich ook in die kamer bevindt. Behalve elektrische bollen, teleporteer-machines en valluiken lopen er ook nog twee soorten vreemde wezens rond. De eerste soort is het gevaarlijkst. Zij kunnen trappen lopen en zich langs glijstangen naar beneden laten zakken. Zij komen in beweging zodra je

langs hun schuilplaats loopt. Zij zijn vast te zetten tussen twee valluiken of lok ze naar de elektrische bollen want ook zij kunnen niet tegen de stralen. Het tweede soort is veel minder gevaarlijk doordat ze minder beweeglijk zijn. Ze komen pas tevoorschijn als hun symbool wordt aangeraakt. Een trap op of af lopen is voldoende om ze kwijt te raken. Ook dit soort kan niet tegen de stralen en valluiken. Schuifdeuren zijn een probleem omdat ze maar acht tellen open blijven, dus na het aanraken van de schakelaar is het rennen geblazen. En of het dan nog niet genoeg is, zijn er ook nog kamers die met een laserkanon worden bewaakt, dat schiet op alles wat beweegt. Na dit alles zijn de lopende banden kinderspel, zij zijn op een eenvoudige manier uit te schakelen of te ontwijken. Ontsnappen is dus niet gemakkelijk. Dit is één van de weinige spelen die met twee spelers kan worden gespeeld, waarbij u elkaar niet tegenwerkt maar juist helpt. Zeker bij het afleiden van de vreemde wezens en het vastzetten van een kanon. Het vinden van voldoende sleutels opent de weg naar buiten.

Bump'n Jump

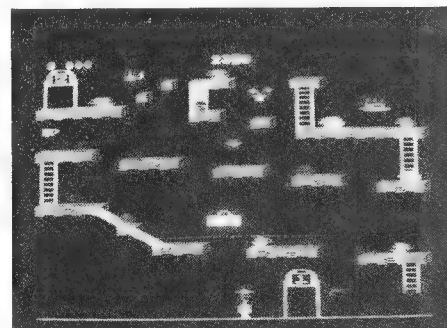
Autoracen is al snel in de vorm van videospelletjes op de buis gebracht met Atari GP en Nintendo-titels als klassiekers. Het is niet makkelijk een keus te maken uit het grote aanbod in dit genre. Bump'n Jump viel op door de wat ruigere aanpak. Bij de meeste spelen is het passeren van een aantal tegenspelers voldoende om het spel te winnen. Bij dit spel is dit niet nodig, de opdracht is: ram ze van de weg, plet ze tegen het asfalt, spring boven op ze, kortom hoe gekker hoe beter. Kijk wel uit, wie een kuil graaft voor een ander, die..... Eenvoudig rondjes rijden is er dus niet bij.

Het lijkt wel of er een orkaan over het circuit heeft gewoed. Alle bruggen zijn verwoest, bermen zijn niet meer recht maar zeer grillig en dus verradelijk. De tegenstanders lijken hier helemaal geen moeite mee te hebben.

Wegversmallingen komen op de vreemdste plaatsen voor en zijn zeer gevaarlijk. Het spel eist veel van het reactievermogen doordat er op de meest verrassende momenten moet worden gesprongen. Tegenstanders hoeft je niet van de weg te rijden, passeren mag ook maar het geheel wordt daardoor wel veel moeilijker. Dit spel blijft op een onverklaarbare manier trekken, terwijl de opzet eenvoudig en zelfs saai is. De aantrekkingskracht ligt vooral in het feit dat je veel agressie in dit spel kwijt kunt.

Miner 2049 / Bounty Bob

Alweer een klassieker, één van de eerste klimspellen. Als mijnwerker speel je Bounty Bob, je opdracht is de mijn te redden van een zekere ondergang. Onderweg kom je natuurlijk allerlei gespuis tegen dat probeert je opdracht te doen mislukken. Iedere kamer waar Bounty Bob komt bestaat uit verschillende verdiepingen, verbonden met ladders, glijbanen en/of liften. Verder is het mogelijk om over niet te grote afstanden heen te springen. Zelfs kun je door je een stukje te laten vallen een (beperkte) afstand overbruggen. Alleen door ervaring is deze afstand te bepalen en door schade en schande wordt men wijs.



Transporteerhokjes zijn ook te gebruiken om je te verplaatsen. De bedoeling is dat Bounty Bob over alle vloerdelen loopt. Dit is te controleren omdat de vloerdelen waar over gelopen is van kleur veranderen. Allerlei vreemde wezens proberen dit te ver-

hinderen, gelukkig staat hij niet geheel machteloos tegen over deze monsters. Er overheen springen kan, maar ook vernietigen is geen probleem. Speciaal voor dit doel bevindt zich in elke kamer mijnwerkersgereedschap. Pak je zo'n stukje gereedschap dan verandert een monster een paar seconden in een vriendelijk mannetje. Op dit moment kun je ze aanraken en verdwijnen ze, zodat je geen last meer van ze hebt. Ook de bloempot met bloemen heeft dit effect, al zie ik dit niet direct als mijnwerkersgereedschap. Het spel wordt gestart met 3 levens, er kunnen geen levens meer worden bijverdiend. Het is dus zaak zeer voorzichtig te werk te gaan.

Jungle Hunt

Na lang gespaard te hebben ga je met je vriendin op safari in de diepste oerwouden van Afrika. Volgens geruchten zijn er zelfs nog kannibalen. Om je vriendin niet te zenuwachtig te maken heb je dat natuurlijk niet verteld. Die mensen kom je toch niet tegen (denk je). Fout natuurlijk, jullie zijn net in het oerwoud als je vriendin wordt gevangen genomen door zo'n stelletje woestelingen. Aangezien je maar één allerliefste vriendin hebt, probeer je haar te redden. Ze is veel te lief en te mooi om als hoofdschotel voor het plaatselijk opperhoofd dienst te doen. Een woeste tocht door het oerwoud is het gevolg. De eerste hindernis is een bijna ondoordringbaar woud. De enige manier om je te verplaatsen is het slingeren van boomtak naar boomtak. Speciaal voor dit doel hangen er overal lianen. Na een paar minuten ben je een volleerde tarzan. De tweede hindernis is al lastiger. Een rivier vol krokodillen en een groot soort krab. Alleen zwemmend is deze hindernis te nemen. Gauw het safarikostuum uit en de weg vervolgen. Krokodillen kun je beter ontwijken door onder ze door te zwemmen. Hou bij dit alles goed de zuurstofvoorraad in de gaten, niets is vervelender dan onder een krokodil zonder zuurstof te zitten.

Na het overwinnen van deze hindernis zien de kannibalen in dat je toch sterker en/of slimmer bent dan ze verwacht hadden. Terwijl jij de heuvel begint te beklimmen gooien zij uit alle macht allerlei stenen naar beneden. De grote vraag blijft open: lukt het je om je vriendin te redden uit de tanden van de kannibalen of vormen jullie samen de hoofdschotel van vanavond??

Beach Head

Een spel dat zeker niet in deze rubriek mag ontbreken is het overbekende

Beach Head. Dat dit spel is aangeslagen blijkt wel uit het feit dat ook beach head 2 is verschenen, beide versies zijn nog volop verkrijgbaar. De opzet van dit spel is een aanvalsmacht aan land te zetten en uiteindelijk een fort op de vijand te veroveren. Jij voert het commando over tien schepen. De eerste keus die gemaakt moet worden, stuur je de schepen door de smalle zeestraat of gaan ze direkt door naar de vijandelijke schepen. Bij de keus voor de zeestraat kom je in een binnenzee bezaaid met zeemijnen, volledig omgeven door rotsen en voorzien van een automatisch afweersysteem dat onafgebroken torpedo's afvuurt. Eén voor één moeten jouw schepen deze hindernissen passeren. Dan wordt er een lucht aanval ingezet. Dit is een zeer realistische gevechtssituatie. De vliegtuigen stijgen op van een vliegdekschip dat buiten schootsveld ligt.

Zij gieren op je af met ratelende boordkannonnen. Denk bij dit spel om het geluisevolume, de burens zullen anders denken dat er een oorlog is uitgebroken en hun bomvrije schuilkeiders opzoeken. Af en toe zou je achter je stoel willen wegkruipen als de kogels op je afkomen, maar daar win je natuurlijk niet mee. Lukt het je om het vliegdekschip uit te schakelen dan beginnen de andere schepen een bombardement. De schepen die deze invasie hebben overleefd kunnen dan aan hun eigenlijke missie beginnen. Zij kunnen hun ladingen aan wal brengen. Elk schip bevat twee tanks. Moeizaam zoeken zij hun weg naar het fort, komen steeds hindernissen tegen zoals muren en anti-tank geschut. Kommen ze uiteindelijk ongeschonden bij het fort aan dan moet dat met 10 welgemikte schoten worden vernietigd. Denk nou niet als je zover bent dat je de overwinning al kunt vieren, want de laatste loodjes wegen het zwaarst.

Quest for Tyres

Met dit spel gaan we terug naar tijden toen de auto nog niet was uitgevonden. Echte helden bewogen zich toen voort op een eenwielige fiets. Je vriendin is geschaakt door een dinosaurus. Dus er snel achteraan om haar uit zijn klauwen te redden.

Onderweg moeten er allerlei hindernissen worden genomen en de dinosaurus geeft haar natuurlijk niet zo maar terug. De eerste hindernis is een grasveld. Geen mooi voetbalveld maar een grasland met vele kuilen. Na enige oefening is het mogelijk over deze kuilen heen te springen. Denk er wel om dat er ook verschillende keien

liggen om het geheel nog wat moeilijker te maken. Als je dit overwonnen hebt kom je in een bosrijk gebied met vele laaghangende takken. Over de omgevalen bomen kan worden heengesprongen en al bukkend moet er onder de takken worden doorgelopen.



De rivier is over te steken door de schildpadden te gebruiken als brug. Het grote probleem is echter dat deze slechts korte tijd boven water blijven, zodat er haast geboden is. Het eist een zekere timing maar het is goed mogelijk. Hou wel rekening met de vervaarlijke dame die met een grote knuppel aan het zwaaien is. Het vervolg valt ook niet mee, maar je mag er ook wel wat voor over hebben om haar te redden. Het is tenslotte je eigen vriendin. Het gaat nu berg op, en blijktbaar zijn ze het er niet mee eens dat je ze zo op de hielen zit. Er worden steeds meer stenen naar beneden gegooid. Bovengekomen is er even tijd voor een pauze. Het wachten is namelijk op een vriendelijke vogel, deze kan je een lift geven. Blijf onder hem rijden en zodra de rivier in zicht is spring je. De vogel brengt je dan naar de overkant. Bij het afdalen van de berg moet je er om denken dat er voldoende snelheid wordt ontwikkeld, zodat je over de volgende rivier kunt springen. Het eind is nu in zicht, een veld dat overgestoken moet worden en een laatste rivier scheiden je van de redding van je vriendin. Lukt dit je zonder dat je gezien wordt dan kom je oog in oog te staan met je vriendin. Wat er dan gebeurt moet je zelf maar ontdekken. Helaas kan je niet al te lang van dit moment genieten want dan begint het spel opnieuw.

StationFall

De knotsgekke robot Floyd is ook weer terug in **StationFall**, een vervolg op **PlanetFall**. Het spelgegeven lijkt veel op een wonderlijk spel verstoppertje. Steve Meretsky's Floyd gaat weer als een robot met verknipt logisch brein te keer en blijft toch altijd ontroerend, voor zover dat op een machine van toepassing kan zijn.

Dankzij het succes in **PlanetFall** werd de held bevorderd van dekschrobber eerste klasse of zo iets naar penneliker subklasse drie of een dergelijke bureaucratische dwarsstraat. In ieder geval is ook het nieuw opgedragen werk stomvervelend. Gelukkig zijn er nog Floyd, jouw voormalige en nu gedegradeerde chef, plus enkele andere robots aan boord. Als spelmaatje neemt u vanzelfsprekend Floyd.

De eerste opdracht, het afleveren van 10 blauwdrukken op het Gamma Delta Gamma 777-G ruimte station lijkt aanvankelijk niet zo'n toer. De SPS Duffy haalt zo'n intergalactisch eindje met gemak. De ontmoeting met de robot Plato, een bejaardere versie van Floyd, aan boord van het ruimtestation moet echter al tot nadenken stemmen. Tussen deze boekenwurm en Floyd ontstaat al gauw de cliché (SF-films) komische robotact met tal van vermakelijke conversaties.

Al spoedig breekt de waanzin los. De hardware slaat op hol. Schuifdeuren blijken ineens levensgevaarlijke doorgangen, androiden maken amok en zelfs Floyd lijkt te desintegreren. Het verhaal laat zich verder moeilijk navertellen en de oplossing geven wij u echt niet.

StationFall komt compleet met bureaucratisch archiefmateriaal, technische handleiding, een Stellar Patrol Patch, paspoort en goede tekstparser. De prijs van dit Infocom/Activision adventures ligt rond de f 100,-.

Hollywood Poker

Strippoker-games behoren tot het soort spelen waar wij niet zo dol op zijn. Niet uit puriteinse overwegingen, maar gewoon omdat het meestal om een duur en kwalitatief slecht "lolletje" gaat. Wie echter van dit soort grappen houdt komt met Robteks **Hollywood Poker** aardig aan zijn trekken. De C-64/C-128-graphics staan voor een redelijke kwaliteit van het ontblootte vrouwelijk schoon. Het is geen Playboy maar de wat krap in het scherm zittende dames kunnen er best mee door. Onderaan het monitorscherm

staan de goed gedetailleerde speelkaarten.

Pokeren tegen Miss Germany en nog vier andere aantrekkelijke pokerpartners kost f 59,- voor de C-64-versie. Info Robtek, Bergen op Zoom, tel.: 01640-56363.



Indiana Jones

Ook de James Bond onder de archeologen, Indiana Jones, moet na het bioscoop- en videosucces van *The Temple of Doom* aan het videogame geloven. US-Gold brengt binnenkort een speciaal op de C-64 afgestemde versie op de markt. Tahir Rashid tekende voor de graphics en dat staat borg voor een goede kwaliteit.

Het labyrinth-deel van het *Temple of Doom*-verhaal leent zich natuurlijk goed voor een arcade-game. Vele nauwe gangen met een doolhofstructuur, tal van vervaarlijke componenten en een met duizelingwekkende vaart rijdend mijnkarretje zijn goed voor heel wat flitsende actie en behendigheidseisen. De spelopdracht om Indiana's vrienden uit de handen van Mola Ram en zijn rituele Thug-moordenaars te redden is dan nog slechts bijzaak.

Onze onverschrokken archeoloog trekt met zijn zweep ten strijde tegen het tempeltuig. Sla de Thugs met de zweep over de rotsen en zij zijn definitief uitgeschakeld. Een flinke klap verdooft hen slechts. Verder is het mogelijk om met behulp van de zweep punten te verdienen door stapels schedels en slangen te raken.

Het eerste spelscherm bestaat uit ladderklommen en het zoeken in de nauwe gangen naar grotten waarin de vijf te redden slavenkinderen gevangen zitten. Op het tweede scherm hebben de Thugs de uitbraak ontdekt en start een wilde achtervolging in mijnkarretjes. Daarbij is het zaak om de juiste weg goed in het oog te houden, want een verkeerde afslag en het karretje schiet echt de "Doom" van de spelers

tegemoet. Tevens moet Indiana proberen zijn karretje achter dat van de Thugs te manoeuvreren en hen vervolgens met de zweep uit te schakelen. Thugs achter je is hoogst omgezond, want deze bezeten killers zijn zwaar bewapend.

Na de achtervolging komt de held in de tempel zelf en moet hij nog even de geheimzinnige Sankara stenen onder het beeld van de moordende god Kali stelen. Daarna gaat de deur naar de vrijheid open. Lukt het drie maal achtereen om de tempel te bereiken en de stenen te stelen dan volgt nog een toetje: Het gevecht op de wankel hangbrug met een horde woeste thugs.

Een aardig arcade actiegame met redelijke graphics. De maalstroom van enerverende gebeurtenissen in de echte Indiana Jones films valt echter niet te evenaren. Toch goed voor heel wat avondjes actieplezier. De prijs voor de C-64/128-versie was op het moment van schrijven nog niet bekend.

Road Runner

Een kruizing tussen een tekenfilm van het geweldadige genre en een speelautomaat. Dat is **Road Runner** waarin de arme Wile E. Coyote probeert de malse Road Runner-vogel te verschalken. Er ontwikkelt zich in de woestijn van Nevada een woest cartoon-videogame met de nodige aanslagen, dirty tricks en listige vallen.

De Road Runner ontwikkelt een gigantische snelheid en is een groot deel van het game slechts als een blauwe schaduw zichtbaar. Wile Coyote is te langzaam om de held zo te vangen, en zal daarom allerlei lage lusten en trucages moeten aanwenden, anders zullen zijn mes en vork voortijdig verroesten. Er zijn 26 van rechts naar links te spelen game-schermen. De oversteek wordt uiteraard bemoeilijkt door de hongerige Coyote die af en toe over een skateboard met raketaandrijving beschikt, mijnenvelden, snelwegen en allerlei obstakels in het landschap. Door het landschap 'runnen' kost een hoop energie die door middel van hoopjes vogelvoedsel aangevuld moet worden. Eet echter niet te veel want, dan wordt u dik, lui en vooral traag. Een toestand die Wile Coyote zo graag ziet. Hetzelfde geldt voor de limonade die een risikante stop, maar wel extra punten oplevert.

Een leuk flitsend arcadespel voor de Commodore 64 met goede graphics,

gladde animaties en cartoon-geluiden zoals Beep, Blurp en Crashes. Dit US Gold-produkt gaat in de winkel ongeveer f 75,- kosten.

Faery Tale

Een sprookje in de traditie van Sword and Sorcery is Microillusion's **Faery Tale**. Een lieflijk stadje, drie ondernemende broers, een magische talisman, tovenarij en een partij afzichtelijke griezels zijn de ingrediënten van dit boeiende Amiga adventure.

Op een nacht wordt de beschermende talisman van het stadje Tambry door een horde duistere figuren, dwergen en dansende skeletten gestolen. Gelukkig heeft de lokale wapenmeester drie zoons die het beschermende talisman zullen heroveren, voordat de boze krachten Tambry in bezit kunnen nemen. Juliaan de oudste broer is de vechtersbaas. Sterk als een os en behendig met de wapens. Philip is de gokker en geluksvogel. Snel van geest en gevat weet hij zich altijd goed door het leven te slaan. Kevin is de dichter en vredelievendste van de drie. Deze jongste broer staat op goede voet met de dierenwereld.

Bij het starten van het game slaat het sprookjesboek open, wordt de geschiedenis van het stadje Tambry verteld en de karakters van de drie broers beschreven. De speler begint als Juliaan. Deze spelfiguur wordt met de muis of joystick door het dorp en in de gebouwen gestuurd. Zodra een gebouw betreden wordt, verschijnt een plattegrond van de vertrekken met de daarin aanwezige voorwerpen. In de gebouwen moeten verschillende objecten verzameld worden. Het gemakkelijkst gaat dat in de herberg waar tegen flinke betaling voedsel en wapens aangeschaft kunnen worden.

Onder het speelscherm staat een stuk perkament waarop de boodschappen van de parser in klassiek schrift verschijnen. Daarnaast staan de muis/joystick-opdrachten Items, Talk, Game, Music, Quit, Magic, Buy Pause, Sound en Load. En rechtsonder een antieke windroos.

Voor de zoekactie zijn een kaart van het eiland Holm (er zijn 17.000 screens), voedsel en wapens onmisbaar. Die wapens zijn hard nodig tegen de woeste dwergen, levende skeletten, spookachtige apen en menseneters. Naast al die engerds zijn er gelukkig ook nog een tovenaars en bedelaars die de held kunnen helpen. Het landschap biedt aardige gedetailleerde semi-3D graphics en is heel afwisselend. De Actie loopt vloeiend zoals wij van de Amiga gewend zijn. Bij de gevechten spat het bloed in het

rond, klinkt realistisch wapengekletter en geschreeuw. Dit alles omlijst door sfeerversterkende muziek.

Slaagt broer Juliaan niet dan wordt Philip in de strijd geworpen en daarna Kevin. Een boeiend Amiga spel voor de liefhebber van sprookjesachtige adventures.

StarBoard2

Extra geheugen en functies aan stuurboord (= de buszijde) van de Amiga? Dat kan met de nieuwe meer-deks **StarBoard2 RAM-kaart** van MicroBotics.

In de eenvoudigste configuratie bestaat de StarBoard2-busmodule uit een nette behuizing met een lege kaart voor maximaal 1 Meg aan Fast RAM. Vult de gebruiker deze kaart met Fast Zero Wait State RAM dan is er nog ruimte voor een extra 1 Meg kaart op dek2 en een multi-function-kaart kaartkaartop dek3.

De multi-functionkaart beschikt over een real-time clock/calender met AAA-batterij, een socket voor de floating point Motorola 68881 FPU, parity checking en een tegen warm rebooten bestendige RAM-disk, de zogenaamde sticky disk. Onmisbaar is het aanwezigge doorgaande busslot voor het koppelen van andere randapparatuur.

prijzen van f800,- tot f2500,-

Flightsimulator II

De wereldberoemde vluchtsimulator **Flightsimulator II** is onlangs ook voor de Commodore Amiga uitgebracht. Van zo'n specifieke Amiga hadden wij wel 68.000-achtige graphics verwacht. Dat valt echter tegen, want de oude Sublogic huisstijl bleef, zij het met een toegenome scherpte en meer aandacht voor details gehandhaafd. Verder niet getreurd, want Flightsimulator II blijft ondanks alles een van de beste micro-vluchtsimulatoren ter wereld. Alleen echt vliegen is nog leuker.

Er komen twee versies, een in kleur en een in monochroom, om het simulatie-effect zo goed mogelijk op de gebruikte monitor over te brengen. Op het moment van testen was de monouitvoering nog niet verkrijgbaar en wij vlogen dus in kleur.

De piloot in spé kan kiezen uit een een-motorige propeller-kist, de Cessna 182, of een moderne business-jet. Een minder interessant onderdeel is de WO-I luchtslag die mank gaat aan de slecht gedetailleerde tegenstanders. Wie echt boeiende gevechten wil kan daarom beter een van de

betere arcade-simulatoren kopen. Veel aantrekkelijker is de optie om via een serieel interface of modem (1200 Baud) samen met een collega-piloot door het zwerk te vliegen.

Er zijn 47 simulatie-scenes. Daarin kan de leerlingpiloot naast het opstijgen en landen ook de navigatie, ramp-situaties en het slecht-weer-vliegen oefenen. De makers weten de realiteit aardig te evenaren. De vrij schetsmatige scenery volgt alle vluchtbewegingen op de voet, een te koude motor slaat onvermijdelijk af, te wilde of onregelmatige capriolen eindigen in een jammerlijke crash en de vlucht verloopt, soms tot vervelens toe, in real time.

Ook is het bijvoorbeeld mogelijk om levensecht tussen de peilers van de Golden Gate-brug van San Francisco te vliegen. Onervaren vliegers kunnen bij deze stunt echter beter een zwemvest meenemen. Verder is er nog een automatische piloot.

Wij konden het niet laten om eens de proef op de som te nemen. Bij de Cessna zitten de bezinnetanks in de vleugel en de motor wat lager in de neus. Dat betekent dat de benzine met hulp van de zwaartekracht van boven naar beneden stroomt. Bij stuntvliegen op de rug of bij een extreme stijghoek krijgt de motor zonder een benzinepomp geen druppel binnen. Toch eens proberen en ja... de machine geeft de pijp aan Maarten. Overigens stond dit feit ook in de handleiding vermeld. Stuntvliegen is onder normale condities niet toegestaan.

Wat onnatuurlijk komt het vliegen met de muis of het keyboard over. Een joystick lijkt immers meer op een stuurknuppel. Dankzij de mogelijkheden van de 68.000 CPU kan het scherm snel in vensters opgedeeld worden of van venster/blik uit de cockpit wisselen. Onderaan het scherm staat het instrumentenvenster waaraan de piloot het wel en wee van zijn of haar kist kan aflezen. De handleiding licht de werking en betekenis van deze instrumenten uitvoerig (in het Engels) toe. Wie onbezorgd wil rondvliegen kan dit deel echter overslaan.

De Sublogic-ontwerpers hebben zich goed van hun taak gekweten. Toch blijven er enige aanmerkingen over. Behalve de licht tegenvallende 3D-graphics zitter er enkele slordigheden in deze simulator. De ILS is hier en daar wat minder nauwkeurig, wij konden enkele vliegvelden niet vinden en de automatische piloot bleek slechts beperkt (boven de 32.000 feet gaat het mis) programmeerbaar.

Niettemin blijft Flightsimulator II een echte topper die het ware vliegen nog

net niet kan vervangen. Sublogic heeft ter aanvulling een hele rees scenery- en simulatie-diskettes aangekondigd. De prijs van deze vluchtsimulator ligt rond de f 180,-.

Het Griekse woord Nemesis staat voor de godin der wrekende gerechtigheid en figuurlijk voor de straf die op de misdaad volgt. Een veel belovende naam voor een misdaadspel, maar het gaat hier om een bekend shoot-them-up arcadegame dat binnenkort onder het Konami-label wordt uitgebracht.

De homecomputerversie van **Nemesis** is natuurlijk niet zo snel als het originele speelhal-game, maar het blijft een van actie flitsende uitdaging. Nemesis speelt, net als bij vele soortgelijke games, in het heelal waar een meedogenloze vijand in de vorm van schepen en monsters de verdedigende speler bedreigt. Zeven niveaus schietwerk, behendig ontwijken en enig strategisch inzicht vereisen het uiterste van de speler. Het game is dan ook veel meer dan de zoveelste upgrade van Space Invaders in deep space.

Na het laden openen zich de verten van de oneindige ruimte. Het blauw is nog rustig, maar daar komt spoedig

verandering. Voor je het weet duiken er hele hordes buitenaardse jagers op de verdedigende destroyer. Na wat oefenwerk lukt het om de eerste golf blauwe tegenstanders af te schieten. Dan verschijnt er onder in beeld een soort menubalk met extra gevechtsmogelijkheden. Er is keuze uit zes iconen: Speed-up, Missiles, Double, Laser, Multiple en een vraagteken. Speed-up (versnellen) spreekt voor zich. De optie Missiles versterkt de defensie met raketten voor het uitschakelen van vijandige basis en/of moederschepen. Double slaat op een extra laserkanon dat onder een hoek van 45 graden allerlei griezels van de bovenste schermrand schiet. Wellicht bedoelden de makers met deze griezels de Nemesis-prophetie? Laser werkt als grof geschut en knalt alles in de vuurlijn naar het hiernamaals. Bij deze geschutskeuze komt het strategie-element om de hoek kijken. De speler kan zijn/haar destroyer niet gewoon afladen met wapentuig en er lekker lukraak op los paffen. Het is een of...of- keuze. Dus of raketten of een extra laserkanon. Een en ander is uiteraard bepalend voor de einduitslag van het komende gevecht.

Het mysterieuze vraagteken toegang tot wat alternatief wapentuig zoals bijvoorbeeld de alles vernietigende Smash-bommen.

In ieder geval blijken de "bommen en granaten" uit de toekomst bijzonder nuttig om de meest vreemde tegenstanders uit te schakelen. Wat dacht u bijvoorbeeld van levende dinosaurus-skeletten, rondzwevende amoeboiden en dodelijk standbeelden die zo van het Paas-eiland zijn weggelopen. Elk scherm brengt weer wat nieuws met als toetje het weerzinwekkende brein dat wacht op de vermetele die zijn hulptscharen durft te vernietigen.

Op de C-64/C-128-computer is het echt aanpopen om de vijand er onder te houden en dat houdt de spanning er goed in. Het spel zal daarom niet gauw vervelen. De graphics zijn weliswaar kleurrijk en fantasievol, maar konden ons inziens nog wel wat beter uitgewerkt worden. De geluidseffecten zijn goed alleen de tune is wat tam. Opmerkelijk is het snelle en gelijkmatige voor- een achteruit scrollen tussen de verschillende schermen.

De prijs van dit Konami game komt vermoedelijk rond de f30,- te liggen.



SETTLE LIGHT SOFT'S DAMMEN

Eindelijk een tegenstander op niveau!

- ★ Nederlandse handleiding met regels en taktische tips
- ★ demonstratie-partijen
- ★ invoeren van zetten met toetsen, cursor of joystick
- ★ terugnemen van vorige zet
- ★ zelf opzetten van standen
- ★ computer speelt zwart of wit
- ★ spiegelen van bestaande stand

In de betere computershop voor

f 37,50 (cassette)

f 45,— (diskette)
incl. BTW

**Ook rechtstreeks te bestellen met
de bestelbon elders in dit blad.**

Zoals je van ons gewend bent staan er in dit nummer weer een groot aantal listings. Zowel langere listings als ook een aantal korte. Het verheugt ons dat ook de commodore 128 en de C16 niet (meer) worden verge-ten. We krijgen hiervoor ook weer wat listings binnen. Er komen er zelfs al een aantal binnen voor de AMIGA. We komen hier zeker binnenkort op terug. Uit het grote aantal listings dat we krijgen ingestuurd blijkt dat er steeds meer wordt gewerkt in machine taal. Op zich kunnen we dit alleen maar toejuichen, de programma's worden er sneller, krachtiger door. Denk er wel om dat de data regels exact zo moeten worden over getypt als dat ze in het blad staan. Er hoeft maar een getal verkeerd te staan en het gehele programma werkt niet meer. Gebruik dus het check-sum programma om dit te controleren. Sterkte en veel plezier met de programma's

R. Goudriaan.

Syntax Checksum

Het overtuiken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.

2. U tikt het RUN commando in. Mocht het program-ma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtuiken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...' komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW com-mando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.

Als alles goed is gegaan loopt er nu een rij regelnum-mers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste program-

ma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit pro-gramma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze lis-tingservice telefoonlijn.

```

1      rem *****
2      rem basic loader "SYNTAX.CHECKSUM"
3      rem runen en new. blijft in ram
        op $c000.
4      rem checksum testen met commando
        : 'sys 49152'
5      rem
6      rem v.840928.64 jan bodzinga
7      rem *****
10     i=49152 :rem beginadres
20     reada:ifa<0then50 :rem data ingel
        ezen
30     pokei,a:i=i+1:goto20
50     poke49184,148:poke49185,192
55     i=49300
60     read a: ifa<0then90
70     pokei,a:i=i+1:goto60
90     print"data[SPACE]is[SPACE]weggezet
        "

95     print"checksum[SPACE]testen[SPACE]
        met[SPACE]sys49152"
100    data 165, 43, 166, 44, 133, 163,
        134, 164, 169, 147
110    data 32, 210, 255, 160, 0, 240,
        3, 32, 73, 192
120    data 32, 73, 192, 208, 1, 96,
        32, 225, 255, 208
130    data 3, 76, 116, 164, 32, 81,
        192, 32, 73, 192
140    data 240, 12, 201, 32, 240, 247,
        24, 101, 167, 133
150    data 167, 76, 37, 192, 166, 167,
        169, 0, 132, 168
160    data 32, 205, 189, 169, 13, 32,
        210, 255, 164, 168
170    data 76, 17, 192, 200, 208, 2,
        230,164,177,163
180    data 96, 162, 0, 189, 123, 192,
        240, 6, 32, 210
190    data 255, 232, 208, 245, 32, 73,
        192, 170, 32, 73
200    data192, 132, 168, 32, 205, 189,
        162, 3, 169, 32
210    data 32, 210, 255, 202, 208, 250,
        169, 0, 133, 167
220    data 164, 168, 96, 82, 69, 71,
        69, 76, 32, 0
230    data -1
240    data 165,197,201,3,240,7,201,4,240
250    data 6,76,148,192,76,34,192,169
260    data 147,32,210,255,76,161,192
270    data -1
    
```

** EINDE LISTING syntax checksum; voor cheksum-
getallen van deze listing z.o.z.

REGEL 1	61	REGEL 130	183	330	print "[HOME] [CRSR-DOWN]":form=1to5
REGEL 2	84	REGEL 140	96		:foro=1to3:gosub30:form=1to6
REGEL 3	6	REGEL 150	96	340	print "[COM H] [4xCRSR-RIGHT]";:next
REGEL 4	117	REGEL 160	127		:print:next:print:next
REGEL 5	143	REGEL 170	71	350	print:print "[2xCRSR-UP]";:gosub30:
REGEL 6	17	REGEL 180	223		forj=1to5:print "[5xCOM Y]";:next
REGEL 7	61	REGEL 190	73	360	x=1113:y=81:z=32
REGEL 10	157	REGEL 200	79	370	forj=1to25:z(j)=z:uu=x+54272:ifz=8
REGEL 20	65	REGEL 210	109		lthenpokeuu+j,0
REGEL 30	203	REGEL 220	106	380	pokex+j,z(j):gosub80:ifj=5orj=10or
REGEL 50	14	REGEL 230	225		j=15orj=20thennext
REGEL 55	251	REGEL 240	114	390	x=x+4:next
REGEL 60	193	REGEL 250	89	400	print "[HOME]":print "[3xCRSR-DOWN] [
REGEL 70	207	REGEL 260	68		10xCRSR-RIGHT]";:forj=1to9:poke646
REGEL 90	245	REGEL 270	224		,6
REGEL 95	237			410	printj;"[2xCRSR-RIGHT]";:ifj=5then
REGEL 100	183				print:print "[3xCRSR-DOWN] [10xCRSR-
REGEL 110	158				RIGHT]";:next
REGEL 120	232			420	next
				430	forj=10to25:printj;"[CRSR-RIGHT]";
				440	ifj=10orj=15orj=20thenprint:print "
					[3xCRSR-DOWN] [10xCRSR-RIGHT]";:nex
					t
				450	next
				460	print:print:t=0
				470	x=1113:y=81:z=32
				480	inputa:ifa=0then730
				490	q=a:gosub40:gosub190
				500	input "[CRSR-UP] [5xCRSR-RIGHT]moet [
					SPACE]over";b=q:b:gosub40:gosub190
				510	ifa>bthengoto540
				520	ifb>athengoto600
				530	q=0:goto40
				540	ifa>6andb<6thenq=0:goto40
				550	c=a-b:q=b-c:ifc=1orc=4orc=5orc=6th
					engoto570
				560	q=0:goto40
				570	gosub220:x=1113:gosub130:q=j:z(j)=
					y:x=x-4:uu=x+54272:pokeuu+j,0:poke
					x+j,z(j)
				580	q=b:x=1113:gosub130:z(j)=z:x=x-4:p
					okex+j,z(j)
				590	q=a:x=1113:gosub130:z(j)=z:x=x-4:p
					okex+j,z(j):goto660
				600	ifb>20anda<20thenq=0:goto40
				610	c=b-a:q=b+c:ifc=1orc=4orc=5orc=6th
					engoto630
				620	q=0:goto40
				630	gosub220:x=1113:gosub130:q=j:z(j)=
					y:x=x-4:uu=x+54272:pokeuu+j,0:poke
					x+j,z(j)
				640	q=a:x=1113:gosub130:z(j)=z:x=x-4:p
					okex+j,z(j)
				650	q=b:x=1113:gosub130:z(j)=32:x=x-4:
					pokex+j,z(j)
				660	yy=40:gosub70:t=t+1:ift=8thengoto6
					80
				670	goto470
				680	pt=0:p=0:forj=1to25:ifz(j)=32thenp
					=p+1:next
				690	ifz(13)=81thenpt=1
				700	ifpt=1andp=24then740
				710	ifj=26then730
				720	next
				730	print "[SHIFT CLR] [3xCRSR-DOWN] [4x
					RSR-RIGHT] jammer!! [SPACE]nr.13[SPA
					CE]is[SPACE]in[SPACE]dit[SPACE]spe
					l[SPACE]uw[SPACE]ge-[4xCRSR-RIGHT]
					luksgetal!":goto1050
				740	print "[SHIFT CLR] [3xCRSR-DOWN] [5x
					RSR-RIGHT]gelukt!!":forj=1to100:po
					ke53280,0:poke53280,1:next:goto105
					0
				750	printspc(8)"[4xCRSR-DOWN] [SPACE]kl

Kleine Solitair

Het volgende programma is er één van Bert Stam uit Zwolle. Hij heeft het programma Kleine Solitair genoemd. Er is een duidelijke uitleg in het programma opgenomen. Laat je echter niet verleiden door de naam, kleine betekent hier zeker niet gemakkelijker. Veel succes bij het spelen van dit verslavende spel.

```

10 print "[SHIFT CLR]":poke53280,11:po
   ke53281,11:poke646,13:gosub 750:go
   to310
20 ifj<1orj>25thengosub60:goto540
30 fork=1to8:print "[CRSR-RIGHT]";:nex
   t:return
40 ifq<1orq>25thenprint "[8xCRSR-RIGHT
   ] [CTRL 9] [SPACE]kan[SPACE]niet!![S
   PACE]":yy=80:q=0:gosub60:goto470
50 return
60 forzz=1to2000:next
70 forz=yytolstep-1:printchr$(20);:ne
   xt:return
80 ifj=5orj=10orj=15orj=20thenx=x+139
90 ifj>5andj<9thenz=y:return
100 ifj>10andj<14thenz=y:return
110 ifj>15andj<19thenz=y:return
120 z=32:return
130 forj=1to25:x=x+4:ifq=jthenreturn
140 ifj=5orj=10orj=15orj=20thenx=x+135
150 next
160 forj=1to25:ifq<>jthennext
170 ifz(j)=32thenreturn
180 q=0:goto40
190 forj=1to25:ifq<>jthennext
200 ifz(j)<>32thenreturn
210 q=0:goto40
220 ifa=6ora=11ora=16ora=21then270
230 ifa=2ora=7ora=12ora=17ora=22then29
   0
240 ifa=10ora=15ora=20then290
250 ifa=4ora=9ora=14ora=19then270
260 return
270 ifb=5orb=10orb=15orb=20thenq=0:got
   o40
280 return
290 ifb=6orb=11orb=16orb=21thenq=0:got
   o40
300 return
310 dimz(25):forj=1to5:gosub30:forl=1t
   o5:print"O[4xCOM Y]";:next:print"[
   COM H]":print
320 form=1to2:print "[CRSR-DOWN]";:next
   :nextj

```

```

eine[2xSPACE]s[SPACE]o[SPACE]l[SPA
CE]i[SPACE]t[SPACE]a[SPACE]i[SPACE
]r[SPACE]e"
760 printspc(18)"[3xCRSR-DOWN]door"
770 printspc(14)"[7xCRSR-DOWN]bert[SPA
CE]stam"
780 printspc(14)"[CRSR-DOWN]anemoonstr
aat[SPACE]27"
790 printspc(14)"[CRSR-DOWN]8012[SPACE
]xv[SPACE] zwolle"
800 forj=1to3500:next
810 print"[SHIFT CLR]":printspc(10)"ee
n[SPACE]korte[SPACE]uitleg!"
820 print"[CRSR-DOWN][3xCRSR-RIGHT]zow
el[SPACE]horizontaal,[SPACE]vertik
aal[SPACE]als"
830 print"[3xCRSR-RIGHT]diagonaal[SPAC
E]moet[SPACE]u[SPACE]met[SPACE]een
[SPACE]balletje"
840 print"[3xCRSR-RIGHT]over[SPACE]een
[SPACE]andere[SPACE]naar[SPACE]een
[SPACE]leeg[SPACE]veld."
850 print"[3xCRSR-RIGHT]het[SPACE]'ges
lagen'[SPACE]balletje[SPACE]wordt[
SPACE]weg-"
860 print"[3xCRSR-RIGHT]genomen.[SPACE
]u[SPACE]kunt[SPACE]niet[SPACE]mee
r[SPACE]dan[SPACE]8"
870 print"[3xCRSR-RIGHT]zetten[SPACE]d
oen.[SPACE]dan[SPACE]blijft[SPACE]
er[SPACE]1[SPACE][CRSR-RIGHT]
880 print"[3xCRSR-RIGHT]balletje[SPACE
]over.[SPACE]het[SPACE]is[SPACE]de
[SPACE]bedoeling"
890 print"[3xCRSR-RIGHT]dat[SPACE]in[S
PACE]het[SPACE]midden[SPACE]een[SP
ACE]balletje"
900 print"[3xCRSR-RIGHT]overblijft![SP
ACE]komt[SPACE]u[SPACE]binnen[SPAC
E]de[SPACE]8"
910 print"[3xCRSR-RIGHT]zetten[SPACE]v
ast[SPACE]te[SPACE]zitten[SPACE]da
n[SPACE]kunt[SPACE]u[SPACE]"

```

```

920 print"[3xCRSR-RIGHT]het[SPACE]spel
[SPACE]stoppen[SPACE]door[SPACE]'0
'[SPACE]te[SPACE]typen"
930 print"[3xCRSR-RIGHT][CTRL 4]een[SP
ACE]voorbeeld:"
940 print"[3xCRSR-RIGHT]u[SPACE]wilt[S
PACE]het[SPACE]balletje[SPACE]in[S
PACE]hokje[SPACE]9"
950 print"[3xCRSR-RIGHT]over[SPACE]hok
je[SPACE]10[SPACE]naar[SPACE]hokje
[SPACE]11.[SPACE]u[SPACE]typt"
960 print"[3xCRSR-RIGHT]eerst[SPACE]9.
[SPACE]als[SPACE]het[SPACE]goed[SP
ACE]is[SPACE]verschijnt"
970 print"[3xCRSR-RIGHT]er[SPACE]'moet
[SPACE]over?'.[SPACE]dan[SPACE]typ
t[SPACE]u[SPACE]10."
980 print"[3xCRSR-RIGHT]de[SPACE]compu
ter[SPACE]doet[SPACE]de[SPACE]rest
!"
990 printspc(12)"[CRSR-DOWN][COM 7][CT
RL 9][SPACE]s[SPACE]u[SPACE]c[SPAC
E]c[SPACE]e[SPACE]s[SPACE]"
1000 printspc(10)"[CRSR-DOWN][COM 1][CT
RL 9][SPACE]druk[SPACE]op[SPACE]re
turn[SPACE]"
1010 getz$:ifz$=""then1010
1020 ifz$=chr$(13)thengoto1040
1030 goto1010
1040 print"[SHIFT CLR]":poke53280,15:po
ke53281,15:poke646,2:goto310
1050 print"[2xCRSR-DOWN][4xCRSR-RIGHT]n
og[SPACE]een[SPACE]keer?[SPACE]{j/
n}"
1060 getz$:ifz$=""then1060
1070 ifz$="n"thenend
1080 clr:print"[SHIFT CLR]":poke53280,1
5:poke53281,15:poke646,2:goto310

```

** EINDE LISTING kleine solitaire **

REGEL	10	238
REGEL	20	193
REGEL	30	126
REGEL	40	161
REGEL	50	142
REGEL	60	58
REGEL	70	165
REGEL	80	57
REGEL	90	116
REGEL	100	204
REGEL	110	214
REGEL	120	57
REGEL	130	122
REGEL	140	53
REGEL	150	130
REGEL	160	166
REGEL	170	204
REGEL	180	90
REGEL	190	166
REGEL	200	126
REGEL	210	90
REGEL	220	9
REGEL	230	228
REGEL	240	47
REGEL	250	227
REGEL	260	142
REGEL	270	202
REGEL	280	142
REGEL	290	206

REGEL	300	142
REGEL	310	157
REGEL	320	172
REGEL	330	121
REGEL	340	26
REGEL	350	153
REGEL	360	41
REGEL	370	2
REGEL	380	127
REGEL	390	252
REGEL	400	47
REGEL	410	112
REGEL	420	130
REGEL	430	221
REGEL	440	171
REGEL	450	130
REGEL	460	220
REGEL	470	41
REGEL	480	239
REGEL	490	208
REGEL	500	228
REGEL	510	136
REGEL	520	133
REGEL	530	90
REGEL	540	142
REGEL	550	213
REGEL	560	90
REGEL	570	103
REGEL	580	158

REGEL	590	252
REGEL	600	230
REGEL	610	209
REGEL	620	90
REGEL	630	103
REGEL	640	157
REGEL	650	169
REGEL	660	54
REGEL	670	36
REGEL	680	70
REGEL	690	227
REGEL	700	107
REGEL	710	48
REGEL	720	130
REGEL	730	190
REGEL	740	3
REGEL	750	140
REGEL	760	124
REGEL	770	234
REGEL	780	103
REGEL	790	120
REGEL	800	214
REGEL	810	255
REGEL	820	208
REGEL	830	63
REGEL	840	103
REGEL	850	140
REGEL	860	102
REGEL	870	179

REGEL	880	112
REGEL	890	81
REGEL	900	87
REGEL	910	45
REGEL	920	77
REGEL	930	135
REGEL	940	46
REGEL	950	11
REGEL	960	154
REGEL	970	255
REGEL	980	64
REGEL	990	146
REGEL	1000	102
REGEL	1010	193
REGEL	1020	44
REGEL	1030	75
REGEL	1040	97
REGEL	1050	173
REGEL	1060	198
REGEL	1070	116
REGEL	1080	55

Powerboot

Deze 'boot' utility kan alleen in combinatie met de power cartridge worden gebruikt. Na het runnen van het programma verschijnt de directory op het scherm. Met als bijzonderheid dat er voor de programma namen getallen verschijnen. Het programma van uw keuze wordt geladen door het indrukken van het daarbij behorende getal.

```

1 rem powerboot / commodore 64
2 rem door r.a. wolswijk
3 rem uit zeist
4 rem
5 pufdef:rem ** boot voor powercart.
6 print"[SHIFT CLR]";:catalog:b=1069
7 :print"[HOME]";:ifpeek(1024)<48the
8 nb=1029:print"[HOME]";
9 dimd(50):t=40:x=1029:c=b-t
10 a=a+1:b=b+t:printstr$(a);":ifpee
11 k(b)=34then8
12 print"[HOME]";:tab(28)"[CTRL 9][SPA
13 CE]welk[SPACE]prgnr.[CTRL 0]"
14 printtab(29);:inputp$:ifp$="~"then
15 run6
16 p=val(p$):ifp<lorp>athen:pufdef:pri
17 nt"[SHIFT CLR]";:new
18 c=c+p*t:fork=1to17:d(k)=peek(c+k):
19 ifpeek(c+k)=34then14
20 next
21 print"[SHIFT CLR]load";chr$(34):fo
22 rl=1to17:ifd(1)=0then16
23 pokex,d(1):x=x+1:next
24 pokex,44:pokex+1,56:poke631,19:pok
25 e632,13:poke633,13
26 poke634,82:poke635,85:poke636,78:p
27 oke637,13:poke198,7:pufdef:end

```

** EINDE LISTING powerboot **

REGEL	1	142	REGEL	10	254
REGEL	2	44	REGEL	11	195
REGEL	3	108	REGEL	12	115
REGEL	4	143	REGEL	13	130
REGEL	5	181	REGEL	14	214
REGEL	6	31	REGEL	15	47
REGEL	7	164	REGEL	16	24
REGEL	8	96	REGEL	17	141
REGEL	9	51			

Tekst

Na het opstarten van dit programma van Jeroen Dobbelaere uit België kunt u met eigen ogen zien wat de mogelijkheden van de commodore 64 zijn in het manipuleren met teksten. De tekst zelf staat in de data regels vanaf regel 1000. U kunt hier ook uw eigen tekst inzetten. Een regel wordt steeds vooraf gegaan door een code die aangeeft hoe een regel op het scherm moet verschijnen. Een uitgebreide uitleg wordt er in het demo gegeven. Een voorbeeld van zo een code is: GR03. G is hierin de code (deze kan variëren van A tot H) R is de richting hier dus rechts en drie is het bepalende cijfer. Laat u niet ontmoedigen, het lijkt allemaal veel moeilijker dan het is. Het programma is erg goed te gebruiken om een eigengemaakt programma te verfraaien.

```

1 rem tekstverfraaiing (cbm/64)
2 rem

```

```

3 rem dobbelaere jeroen
4 rem
5 rem 9400ninove (belgie)
6 rem
7 rem
10 poke53280,0:poke53281,0:poke646,1:
11 print"[SHIFT CLR]"
12 reada$
13 b$=left$(a$,1):a$=right$(a$,len(a$)-1)
14 ifb$="?"thenprinta$:goto20
15 ifb$="s"thenend
16 ifb$="a"then100
17 ifb$="b"then200
18 ifb$="c"then300
19 ifb$="d"then400
20 ifb$="e"then500
21 ifb$="f"then600
22 ifb$="g"then700
23 ifb$="p"thenform=1to500:next:goto2
24 0
25 ifb$="h"then800
26 print"error":end
27 form=1tolen(a$)
28 print"[CRSR-DOWN][CRSR-LEFT][SPACE]
29 [COM Y][CRSR-UP][CRSR-LEFT]"mid$(
30 a$,n,1);:next
31 form=1to800:next:print"[CRSR-DOWN]
32 [CRSR-LEFT][SPACE][CRSR-UP]";:goto2
33 0
34 gosub900
35 form=1to19:printtab(20-n)left$(a$,
36 n)right$(a$,n)"[CRSR-UP]";:next:pri
37 nt
38 form=1to500:next:goto20
39 gosub900
40 form=1to38:printright$(a$,n)"[CRSR
41 -UP]";:next:print:form=1to800:next:
42 goto20
43 gosub900
44 form=1to38:printtab(39-n)left$(a$,
45 n)"[CRSR-UP]";:next:print:form=1to8
46 00:next:goto20
47 v=val(a$)
48 form=1tov
49 form=1to20:print"[SHIFT INST][SPAC
50 E]"tab(20)chr$(20)chr$(20)"[CRSR-U
51 P]";:next:print
52 next:goto20
53 form=1tolen(a$):b$=mid$(a$,len(a$)+1-n,1)
54 print"[HOME]";:form=1to23:print"[S
55 PACE]";:printb$[CRSR-UP]";:next
56 print"[SHIFT INST]";:next:print"[2x
57 CRSR-DOWN][2xCRSR-UP]"
58 goto20
59 b$=left$(a$,3):b=val(right$(b$,2))
60 b$=left$(b$,1):a$=mid$(a$,4)
61 c$="[40xSPACE]"
62 a$=c$+a$+c$:s=1:ifb$="r"thens=-1
63 l1=1:l2=len(a$)-39:ifs=-1thenl1=l2:l2=1
64 fora=1tob
65 form=1to12steps:foraa=0to9:next
66 printmid$(a$,n,38)"[CRSR-UP]";:next
67 n,a
68 goto20
69 gosub900
70 form=1to38:form=1ton:ifmid$(a$,n,1)
71 ="[SPACE]"then830
72 print"[SPACE]"mid$(a$,n,1)"[CRSR-L
73 EFT]";:fora=1to2:nexta,m
74 print"[SPACE]";:printtab(n)mid$(a$,
75 n,1)"[2xCRSR-UP]";:next:print:print

```

```

:goto20
900  iflen(a$)<37thena$="[SPACE]"+"a$+"[
    SPACE]":goto900
905  iflen(a$)<38thena$=a$+"[SPACE]"
910  return
1000  data"?[SHIFT CLR][4xCRSR-DOWN]"
1010  data"adit[SPACE]is[SPACE]een[SPACE]
    voorbeeld[SPACE]van[SPACE]enkele[
    SPACE]"
1020  data"asoorten[SPACE]van[SPACE]teks
    t[SPACE]opkomen."
1030  data"?[3xCRSR-DOWN]"
1040  data"bje[SPACE]kan[SPACE]bijvoorbe
    eld[SPACE]zo[SPACE]opkomen..."?"[
    2xCRSR-UP]"
1050  data"ben[SPACE]er[SPACE]dan[SPACE]
    nog[SPACE]eens[SPACE]tussenkomen"
1060  data"bof[SPACE]er[SPACE]gewoon[SPA
    CE]onder[SPACE]komen","?[CRSR-DOWN
    ]
1070  data"cje[SPACE]kan[SPACE]ook[SPACE]
    van[SPACE]links[SPACE]komen"
1080  data"den[SPACE]dus[SPACE]ook[SPACE]
    van[SPACE]rechts..."
1090  data"a[SPACE]en[SPACE]regels[SPACE]
    wegdoen[SPACE]is[SPACE]niets,[SPA
    CE]hier[SPACE]mee"
1100  data"bkijk[SPACE]maar[SPACE]:","?[
    HOME][4xCRSR-DOWN]"
1110  data"el6","?[SHIFT CLR]"
1130  data"fooit[SPACE]al[SPACE]eens[SPA
    CE]vallende[SPACE]letters[SPACE]ge
    zien"
1140  data"fnog[SPACE]niet[SPACE]?,[SPAC
    E]dan[SPACE]zie[SPACE]je[SPACE]ze[
    SPACE]nu[SPACE]!!!"
1150  data"fdat[SPACE]is[SPACE]eens[SPAC
    E]iets[SPACE]anders,[SPACE]hee..."
1160  data"gl04doorlopende[SPACE]tekst[S
    PACE]gaat[SPACE]hier[SPACE]ook[SPA
    CE]mee,[SPACE]zoals[SPACE]je[SPACE]
    hier[SPACE]ziet..."
1170  data"gr04en[SPACE]je[SPACE]kan[SPA
    CE]er[SPACE]ook[SPACE]links[SPACE]
    en[SPACE]rechts[SPACE]mee[SPACE]ga
    an..."?"[2xCRSR-DOWN][CRSR-UP]"
1180  data"gr0ldit[SPACE]gaat[SPACE]dus[
    SPACE]nu[SPACE]naar[SPACE]rechts"
1190  data"gl0len[SPACE]dit[SPACE]dus[SP
    ACE]naar[SPACE]links[SPACE]weer"
1200  data"gr0ldit[SPACE]gaat[SPACE]dus[
    SPACE]nu[SPACE]naar[SPACE]rechts"
1210  data"gl0len[SPACE]dit[SPACE]dus[SP
    ACE]naar[SPACE]links[SPACE]weer"
1220  data"p."
1230  data"hldit[SPACE]lijkt[SPACE]op[SPA
    CE]vallende[SPACE]letters,"
1240  data"hmaar[SPACE]het[SPACE]is[SPAC
    E]het[SPACE]niet[SPACE]!!!!"
1250  data"p."?"[SHIFT CLR]"
1260  data"anu[SPACE]nog[SPACE]eens[SPAC
    E]alles[SPACE]op[SPACE]een[SPACE]r
    ijtje[SPACE]:"
1270  data"ahet[SPACE]werkt[SPACE]met[SP
    ACE]data,[SPACE]en[SPACE]je[SPACE]
    moet[SPACE]voor[SPACE]je"
1280  data"azin[SPACE]telkens[SPACE]een[
    SPACE]code[SPACE]zetten"
1289  data"adit[SPACE]wordt[SPACE]bijvoo
    rbeeld[SPACE]:"
1290  data"aadit[SPACE]wordt[SPACE]bijvo
    orbeeld[SPACE]:"
1300  data"anu[SPACE]de[SPACE]effecten[S
    PACE]met[SPACE]code","aen[SPACE]ui

```

```

tleg[SPACE]als[SPACE]nodig."
1310  data"ap.[SPACE]:[SPACE]pause","p."
1320  data"aa[SPACE]dient[SPACE]hier[SPA
    CE]voor."
1330  data"bb[SPACE]dient[SPACE]hier[SPA
    CE]voor."
1340  data"cc[SPACE]dient[SPACE]hier[SPA
    CE]voor."
1350  data"dd[SPACE]dient[SPACE]hier[SPA
    CE]voor."
1360  data"ae[SPACE]+aantal[SPACE]:[SPAC
    E]regels[SPACE]wegdoen"
1370  data"a[SPACE]voorbeeld[SPACE]:","?
    [HOME]"
1380  data"el8","?[SHIFT CLR]"
1390  data"ff[SPACE]dient[SPACE]hier[SPA
    CE]voor."
1400  data"ag+richting+aantal([SPACE]in[
    SPACE]twee[SPACE]cijfers[SPACE])"
1410  data"a[SPACE]voorbeeld:[SPACE]gl03
    ***[SPACE]commodore[SPACE]info[SP
    ACE]****"
1420  data"gl03***[SPACE]commodore[SPAC
    E]info[SPACE]****"
1430  data"ahetzelfde,[SPACE]maar[SPACE]
    l[SPACE]wordt[SPACE]r"
1440  data"gr03***[SPACE]commodore[SPAC
    E]info[SPACE]****"
1450  data"hh[SPACE]dient[SPACE]hier[SPA
    CE]voor"
1460  data"hverder[SPACE]hebben[SPACE]we
    [SPACE]nog[SPACE]?[SPACE](print)"
1470  data"cen[SPACE]s(top)"
1480  data"s."

```

** EINDE LISTING tekst **

REGEL	1	4	REGEL	530	167
REGEL	2	143	REGEL	600	119
REGEL	3	23	REGEL	610	0
REGEL	4	60	REGEL	620	136
REGEL	5	36	REGEL	630	235
REGEL	6	250	REGEL	700	23
REGEL	7	143	REGEL	710	37
REGEL	10	157	REGEL	720	93
REGEL	20	236	REGEL	730	9
REGEL	30	68	REGEL	731	239
REGEL	40	240	REGEL	740	139
REGEL	50	97	REGEL	750	212
REGEL	60	96	REGEL	760	118
REGEL	70	98	REGEL	770	235
REGEL	80	100	REGEL	800	38
REGEL	90	102	REGEL	810	242
REGEL	91	104	REGEL	820	123
REGEL	92	106	REGEL	830	14
REGEL	93	108	REGEL	900	124
REGEL	94	170	REGEL	905	51
REGEL	95	110	REGEL	910	142
REGEL	96	33	REGEL	1000	221
REGEL	100	207	REGEL	1010	152
REGEL	110	190	REGEL	1020	233
REGEL	120	37	REGEL	1030	57
REGEL	200	38	REGEL	1040	6
REGEL	210	15	REGEL	1050	113
REGEL	220	204	REGEL	1060	148
REGEL	300	38	REGEL	1070	60
REGEL	310	250	REGEL	1080	171
REGEL	400	38	REGEL	1090	176
REGEL	410	42	REGEL	1100	147
REGEL	500	131	REGEL	1110	181
REGEL	510	172	REGEL	1130	48
REGEL	520	232	REGEL	1140	123

REGEL 1150	39	REGEL 1340	93	430	ifsb=119then470
REGEL 1160	187	REGEL 1350	95	440	goto490
REGEL 1170	247	REGEL 1360	46	450	bb=bb-hs:ifbb<18thenbb=18
REGEL 1180	57	REGEL 1370	166	460	goto480
REGEL 1190	241	REGEL 1380	183	470	bb=bb+hs:ifbb>66thenbb=66
REGEL 1200	57	REGEL 1390	99	480	poke53254,bb
REGEL 1210	241	REGEL 1400	209	490	rem bal 2
REGEL 1220	69	REGEL 1410	251	500	onrbgoto510,570,600,510
REGEL 1230	167	REGEL 1420	222		vb=vb+bs:poke53251,vb:ifvb>dbthen5
REGEL 1240	99	REGEL 1430	30		30
REGEL 1250	135	REGEL 1440	228	520	goto630
REGEL 1260	164	REGEL 1450	57	530	ifbb<hb-11then690
REGEL 1270	206	REGEL 1460	24	540	ifbb>hb+3then690
REGEL 1280	220	REGEL 1470	52	550	rb=2:poke54273,21:poke54272,31:pok
REGEL 1289	42	REGEL 1480	7		e54276,17:forp=1to100:next:poke542
REGEL 1290	107				76,16
REGEL 1300	2			560	goto630
REGEL 1310	44			570	vb=vb-bs:poke53251,vb
REGEL 1320	89			580	ifvb<75thenrb=3:bs=bs*1.05:poke532
REGEL 1330	91				51,0

Duo-bal

Wim Stadhouders, een van onze regelmatige inzenders van programma's en kenners weten dat dit altijd doordachte programma's zijn, heeft voor ons een spel gemaakt dat hij Duo-bal heeft genoemd. Er bevinden zich, zowel links als rechts op het scherm, vier kolommen met daarin ballen. Deze komen willekeurig naar beneden. Door het verplaatsen van een slaghout moet u de ballen zien tegen te houden. Het spel is verloren als u van een groep vier ballen niet tijdig heeft teruggeslagen.

```

100 rem *** duo-bal
110 rem *** door w.a. stadhouders
120 rem ***
130 rem *** terneuzen
140 goto940
150 rem lus
160 sc=int(ti/60):print"[HOME][COM 4]"
    sc
170 rem linker blok
180 sa=peek(56321)
190 ifsa=251then220
200 ifsa=247then240
210 goto260
220 ba=ba-hs:ifba<28thenba=28
230 goto250
240 ba=ba+hs:ifba>76thenba=76
250 poke53252,ba
260 rem bal verplaatsen
270 onragoto280,340,370
280 va=va+as:poke53249,va:ifva>dathen3
    00
290 goto400
300 ifba<ha-11then640
310 ifba>ha+3then640
320 ra=2:poke54273,16:poke54272,195:po
    ke54276,17:forp=1to100:next:poke54
    276,16
330 goto400
340 va=va-as:poke53249,va
350 ifva<75thenra=3:as=as*1.05:poke532
    49,0
360 goto400
370 rem opnieuw
380 wa=int(4*rnd(1)):ha=32+wa*16:ra=1
390 va=74:poke53248,ha:poke53249,va
400 rem rechter blok
410 sb=peek(56320)
420 ifsb=123then450

```

```

590 goto630
600 rem opnieuw
610 wb=int(4*rnd(1)):hb=24+wb*16:rb=1
620 vb=74:poke53250,hb:poke53251,vb
630 goto150
640 rem links mist
650 ra=3:fa=fa-1
660 gosub740
670 poke1145+fa*2,32:iffa=0then770
680 poke53249,0:goto150
690 rem rechts mist
700 rb=3:fb=fb-1
710 gosub740
720 poke1176+fb*2,32:iffb=0then770
730 poke53251,0:goto150
740 poke53280,11:poke54273,28:poke5427
    2,49:poke54276,33:forp=1to400:next
    poke54273,14:poke54272,24:forp=1to
    800:next:poke54276,32
760 forp=1to800:next:poke53280,5:retur
    n
770 rem einde spel
780 poke53281,10:poke54273,16:poke5427
    2,195:poke54276,17
790 print"[HOME][CRSR-DOWN][CTRL 2][CT
    RL 9]<<<[SPACE]einde[SPACE]spel[SP
    ACE]...[SPACE]>>>"
800 forp=1to800:next:poke54273,18:forp
    =1to800:next:poke54273,16
810 forp=1to800:next:poke54273,19:forp
    =1to1600:next:poke54276,16:poke532
    81,13
820 forp=1to1000:next:print"[SHIFT CLR
    ]":poke53269,0
830 geta$:ifa$=""then850
840 goto830
850 print"[CTRL 6]wilt[SPACE]u:"
860 print"[CRSR-DOWN]1.[SPACE]een[SPAC
    E]spel[SPACE]met[SPACE]andere[SPAC
    E]condities"
870 print"2.[SPACE]nog[SPACE]een[SPACE
    ]spel[SPACE]met[SPACE]dezelfde[SPA
    CE]condities"
880 print"3.[SPACE]stoppen."
890 inputk$
900 ifk$="1"then1150
910 ifk$="2"then1240
920 ifk$="3"thensys65126:end
930 goto890
940 print"[SHIFT CLR]":poke53281,13:po
    ke53280,5
950 forp=832to894:pokep,0:next:poke832
    ,60:poke835,126:poke838,255:poke84
    1,255

```



```

960 poke844,255:poke847,255:poke850,12
    6:poke853,60
970 poke2040,13:poke2041,13:poke53287,
    6:poke53288,9
980 forp 896to958:pokep,0:next:forp=89
    6to917step3:pokep,255:next
990 poke53277,12:poke53271,0:poke2042,
    14:poke2043,14:poke53264,10
1000 poke53289,6:poke53290,9
1010 forp=54272to54296:pokep,0:next:pok
    e54278,72:poke54296,15
1020 print"[SHIFT CLR][CRSR-DOWN][COM 4
    ][9xSPACE]<[SPACE]d[SPACE]u[SPACE]
    o[SPACE]-[SPACE]b[SPACE]a[SPACE]l[
    SPACE]>"
1030 print"[CRSR-DOWN]zowel[SPACE]links
    [SPACE]als[2xSPACE]rechts[SPACE]op
    [SPACE]het[SPACE]scherm"
1040 print"bevinden[SPACE]zich[SPACE]vi
    er[SPACE]kolommen[SPACE]met[SPACE]
    daarin"
1050 print"ballen."
1060 print"deze[SPACE]komen[SPACE]wille
    keurig[SPACE]naar[SPACE]beneden."
1070 print"door[SPACE]het[SPACE]verplaa
    tsen[SPACE]van[SPACE]een[SPACE]sla
    ghout"
1080 print"moet[SPACE]u[SPACE]de[SPACE]
    ballen[SPACE]zien[SPACE]tegen[SPAC
    E]te[SPACE]houden,"
1090 print"zodat[SPACE]ze[SPACE]weer[SP
    ACE]naar[SPACE]boven[SPACE]gaan."
1100 print"u[SPACE]hebt[SPACE]verloren,
    [SPACE]als[SPACE]u[SPACE]van[SPACE]
    een[SPACE]groep"
1110 print"vier[SPACE]ballen[SPACE]niet
    [SPACE]tijdig[SPACE]hebt[SPACE]ter
    ug-"
1120 print"geslagen."
1130 print"[CRSR-DOWN]<[SPACE]druk[SPAC
    E]op[SPACE]een[SPACE]toets[SPACE]o
    m[SPACE]verder[SPACE]te[SPACE]gaan
    [SPACE]>"
1140 geta$:ifa$=""then1140
1150 rem keuzes
1160 print"[SHIFT CLR][CTRL 7][CRSR-DOW

```

```

N][6xSPACE]<[SPACE]keuze-mogelijkh
eden[SPACE]>"
1170 print"[CRSR-DOWN]de[SPACE]beginsne
    lheid,[SPACE]waarmee[SPACE]de[SPAC
    E]ballen[SPACE]naarbeneden[SPACE]k
    omen[SPACE](1-8)"
1180 inputqs:ifqs<lorqs>8then1180
1190 as=qs:bs=qs*1.1
1200 print"[CRSR-DOWN]de[SPACE]snelheid
    ,[SPACE]waarmee[SPACE]u[SPACE]het[
    SPACE]slaghout[SPACE]ver-plaatst[S
    PACE](1-4)"
1210 inpuths:ifhs<lorhs>4then1210
1220 print"[CRSR-DOWN]de[SPACE]hoogte,[
    SPACE]waarover[SPACE]de[SPACE]bal[
    SPACE]valt[SPACE](5-18)"
1230 inputvh:ifvh<5orvh>18then1230
1240 rem start spel
1250 as=qs:bs=qs*1.1
1260 print"[SHIFT CLR][5xSPACE]score"
1270 print"[CTRL 6][CRSR-DOWN][CTRL 9][
    40xSPACE]";
1280 print"[COM L][CTRL 0][CTRL 1]W[SHI
    FT -]W[SHIFT -]W[SHIFT -]W[SHIFT -
    ][22xSPACE][SHIFT -]W[SHIFT -]W[SH
    IFT -]W[SHIFT -]W[CTRL 9][CTRL 6][
    COM J]";
1290 forp=1tovh:print"[CTRL 9][COM L][C
    CTRL 0][CTRL 1][SPACE][SHIFT -][SHI
    FT SPACE][SHIFT -][SHIFT SPACE][SH
    IFT -][SHIFT SPACE][SHIFT -][22xSP
    ACE][SHIFT -][SHIFT SPACE][SHIFT -
    ][SHIFT SPACE][SHIFT -][SHIFT SPAC
    E][SHIFT -][SHIFT SPACE][CTRL 9][C
    CTRL 6][COM J]";:next
1300 rem variabelen
1310 ra=3:fa=4:da=76+vh*8:ba=28:poke532
    52,ba:p=84+vh*8:poke53253,p
1320 rb=3:fb=4:db=da:bb=18:poke53254,bb
    :poke53255,p
1330 poke53249,0:poke53251,0:poke53269,
    15:ti$="000000"
1340 goto150

```

** EINDE LISTING duo-bal **

REGEL	100	241	REGEL	360	29	REGEL	620	208	REGEL	880	149
REGEL	110	123	REGEL	370	182	REGEL	630	31	REGEL	890	244
REGEL	120	76	REGEL	380	76	REGEL	640	77	REGEL	900	143
REGEL	130	36	REGEL	390	219	REGEL	650	78	REGEL	910	144
REGEL	140	38	REGEL	400	196	REGEL	660	40	REGEL	920	38
REGEL	150	131	REGEL	410	90	REGEL	670	117	REGEL	930	42
REGEL	160	85	REGEL	420	168	REGEL	680	83	REGEL	940	8
REGEL	170	124	REGEL	430	175	REGEL	690	149	REGEL	950	74
REGEL	180	90	REGEL	440	38	REGEL	700	81	REGEL	960	113
REGEL	190	164	REGEL	450	171	REGEL	710	40	REGEL	970	145
REGEL	200	171	REGEL	460	37	REGEL	720	123	REGEL	980	215
REGEL	210	33	REGEL	470	174	REGEL	730	76	REGEL	990	20
REGEL	220	169	REGEL	480	74	REGEL	740	78			
REGEL	230	32	REGEL	490	144	REGEL	750	228	REGEL	1000	61
REGEL	240	172	REGEL	500	206	REGEL	760	168	REGEL	1010	75
REGEL	250	71	REGEL	510	137	REGEL	770	40	REGEL	1020	118
REGEL	260	163	REGEL	520	34	REGEL	780	157	REGEL	1030	235
REGEL	270	208	REGEL	530	159	REGEL	790	169	REGEL	1040	131
REGEL	280	133	REGEL	540	109	REGEL	800	102	REGEL	1050	185
REGEL	290	29	REGEL	550	250	REGEL	810	253	REGEL	1060	4
REGEL	300	152	REGEL	560	34	REGEL	820	181	REGEL	1070	91
REGEL	310	102	REGEL	570	183	REGEL	830	106	REGEL	1080	207
REGEL	320	56	REGEL	580	21	REGEL	840	36	REGEL	1090	18
REGEL	330	29	REGEL	590	34	REGEL	850	202	REGEL	1100	93
REGEL	340	186	REGEL	600	182	REGEL	860	144	REGEL	1110	131
REGEL	350	24	REGEL	610	81	REGEL	870	248	REGEL	1120	81

REGEL	1130	184	REGEL	1240	81	39	dataf72093c0a2008e04d4a2118e04d4a6
REGEL	1140	147	REGEL	1250	75		fca4fd2082c08d45c28e46c2a956
REGEL	1150	102	REGEL	1260	236	40	data8dcd05a21fa0e4c8d0fde8d0f84cec
REGEL	1160	129	REGEL	1270	89		c0ee20d0ad00dcc96fd0f64c00c0
REGEL	1170	1	REGEL	1280	117	41	data124452554b20565555524b4e4f5000
REGEL	1180	36	REGEL	1290	27		53434f52453a2020202020202000
REGEL	1190	75	REGEL	1300	104	42	data0103052a2b2c51525354557a7b7ca1
REGEL	1200	164	REGEL	1310	166		a3a500000000000000000000000000
REGEL	1210	255	REGEL	1320	198		
REGEL	1220	123	REGEL	1330	164		
REGEL	1230	67	REGEL	1340	3		

Explosieven

Het spel explosieven, gemaakt door Thom Vreven uit Bennekom, is een aktiespel. Het wordt gespeeld met de joystick in poort twee. Het is de bedoeling zoveel mogelijk tijdbommen onschadelijk te maken voordat ze exploderen. Per bom die ontploft verschijnen er twee nieuwe voor in de plaats en voor elke geruimde bom verschijnt er een nieuwe. Het spel is afgelopen als men te dicht bij een ontplofende bom is. Per bom krijgt men punten, afhankelijk van de tijd die de bom aan geeft. Het programma is in M.C. geschreven en is door een basicloader op z'n plaats gezet.

```

1      rem explosieven / thom vreven
10     forr=0to22:reada$:forb=1tolen(a$)s
      tep2:c$=mid$(a$,b,1):gosub12:c=16*
      g
11     c$=mid$(a$,b+1,1):gosub12:c=c+g:po
      ke49152+t,c:t=t+1:nextb,r:sys49152
12     g=asc(c$)-48:ifg>9theng=g-7
13     return
20     data2044e5a200a90e9d00d89d00d99d00
      da9d00dbe8d0f1a9a09d0004e8e0
21     data78d0f8a2009dc007e8e028d0f8a200
      8e20d08e21d0189da8069d9f049d
22     data78049d90059db7059dcf068a6928b0
      06aaa9a04c33c0a2078e8602a201
23     dataaa00d200ce5a0c2a961201eabae00dc
      e06fd0f9a90f8d18d4a9018d00d4
24     data8d06d4a90a8d01d48d05d44cc1c08a
      a204c888f00918692890f8e84c86
25     datac060a6a2bd00c0186911b0034c99c0
      6904a8e6a2bd00c1186922b0034c
26     dataa8c069022083c08dbec08ebfc0a939
      8de90560a20d86f72093c0e6fbc6
27     dataf7d0f7a201a00d200ce5a0c2a97020
      1eaba21486f9a20a86fc86faa200
28     data86f786f8a201a014200ce5a6f7a5f8
      20cdbc6fba4fbc005d035a200bd
29     data0004c966d005a9209d0004bd0005c9
      66d005a9209d0005bd0006c966d0
30     data05a9209d0006bd0007c966d005a920
      9d0007e8d0cdc00f0034ca5cla2
31     data0a86fba2048e59c18e4ecla200bd00
      07c9319007c93ab003de0007c930
32     data03486fea0008c04d4a0818c04d4ac
      4ec18a18e952b001888c84c18d83
33     datac1a000b97ec2aaa9669d7205c8c011
      d0f22093c02093c0a6fee8d0b7ac
34     data4ec1c007f009ee4eclee59c14c4ac1
      a6f9a4faad00dc85fe2901d00188
35     dataa5fe2902d001c8a5fe2904d001caa5
      fe2908d001e8c002d001c8c018d0
36     data0188e000d001e8e027d001ca86fc84
      fda6f9a4fa2082c08df4c18ef5c1
37     data8d00c28e01c2accd05c066d0034c5a
      c2a0208ccd05a6fc86f9a4fd84fa
38     data2082c08d14c28e15c2adcd05c93090
      1bc93ab017e92f65f79002e6f885

```

** EINDE LISTING explosieven **

REGEL	1	30	REGEL	29	247
REGEL	10	12	REGEL	30	45
REGEL	11	215	REGEL	31	78
REGEL	12	77	REGEL	32	109
REGEL	13	142	REGEL	33	149
REGEL	20	135	REGEL	34	211
REGEL	21	120	REGEL	35	101
REGEL	22	74	REGEL	36	166
REGEL	23	166	REGEL	37	200
REGEL	24	121	REGEL	38	117
REGEL	25	38	REGEL	39	116
REGEL	26	168	REGEL	40	23
REGEL	27	131	REGEL	41	102
REGEL	28	181	REGEL	42	87

Files Uitprinten

Het programma files uitprinten is gebaseerd op een oud thema. De maker E.J. Lubberts uit Deventer ergerde zich aan het vele verspillen van papier bij het uitprinten van directories. Hij heeft daarom een programma geschreven dat de keuze geeft deze directories in een, twee of in drie kolommen uit te printen. Er kan een keuze gemaakt worden door middel van de funktietoetsen.

```

1      rem *****
2      rem * files uitprinten in *
3      rem * 1,2 of 3 kolommen *
4      rem * e.j.lubberts *
5      rem *
6      rem * deventer *
7      rem *****
10     poke53280,1:poke53281,1:printchr$(
      30)chr$(14)chr$(147)
20     print:print
30     print"[CTRL 9][14xSPACE]file[SPACE
      ]printer[14xSPACE]"
40     print:print:print
50     print"Maak [SPACE]Uw[SPACE]keuze."
60     print:print:print
70     print"f1[SPACE]=[SPACE]directory[S
      PACE]op[SPACE]het[SPACE]scherm[SPA
      CE]laten[SPACE]zien."
80     print
90     print"f3[SPACE]=[SPACE]directory[S
      PACE]in[SPACE]1[SPACE]kolom[SPACE]
      uitprinten."
100    print:print
110    print"f5[SPACE]=[SPACE]directory[S
      PACE]in[SPACE]2[SPACE]kolommen[SPA
      CE]uitprinten."
120    print
130    print"f7[SPACE]=[SPACE]directory[S
      PACE]in[SPACE]3[SPACE]kolommen[SPA
      CE]uitprinten."
140    poke198,0:wait198,1
150    getc$:ifc$=chr$(133)then200
160    ifc$=chr$(134)thengosub300:gosub60
      0
170    ifc$=chr$(135)thengosub300:gosub80

```

```

0
180 ifc$=chr$(136) then gosub 300: gosub 10
    00
185 ifc$<>chr$(133) or c$<>chr$(134) or c$
    <>chr$(135) or c$<>chr$(136) then 10
190 goto 150
200 rem *****
210 rem * dj. op het scherm *
220 rem *****
230 print "[SHIFT CLR]": open 1, 8, 0, "$": p
    oke 781, 1: sys 65478: geta$, a$
240 geta$, a$: ifst=64 then sys 65484: close
    1: goto 275
250 geta$, b$: print "[CURSR-LEFT]" asc(a$+
    chr$(0))+256*asc(b$+chr$(0));
260 geta$: print a$: ifa$<>" " then 260
270 print: goto 240
275 print: print "[7xSPACE] [CTRL 9] [SPAC
    E] Druk [SPACE] op [SPACE] een [SPACE] to
    ets [SPACE] [CTRL 0]"
280 poke 198, 0: wait 198, 1: goto 10
290 :
300 rem *****
    **
310 rem * dj. op vragen voor uitprinten
    *
320 rem *****
    **
330 dim f$(152)
340 dim g$(152)
350 print chr$(147) chr$(13) chr$(142)
360 open 9, 8, 0, "$"
370 get #9, a$, b$: n0=0
380 get #9, a$, b$: z0=0: x$=""
390 ifst<>0 then 470
400 get #9, a$, b$
410 n$=chr$(0)
420 g$=right$(" [6xSPACE]" + str$(asc(a$+
    n$)-asc(b$+n$)*256), 6)
430 get #9, a$: ifa$=chr$(34) then z0=sgn(z
    0+1)
440 ifa$="" then n0=n0+1: n1=n0-1: gosub 48
    0: goto 380
450 ifz0=1 then g$=g$+a$
460 goto 430
470 close 9: goto 590
480 ifn0=1 then 560
490 if len(g$)=6 then 580
500 aa$="" : for i=8 to 24: c$=mid$(g$, i, 1):
    ifc$=chr$(34) then i=24: goto 520
    aa$=aa$+c$
510 next i: g$(n1)=left$(aa$+" [19xSPACE]
    ", 18)+left$(g$, 6)+" [2xSPACE]" + id$
520 h1$=mid$(g$(n1), 22, 3): h2$=left$(g$
    (n1), 16)
530 h$=h1$+" [SPACE]" + h2$+" [SPACE]" + mid
    $(g$, 26, 3): print "[7xSPACE]" h$: a=a+
    1: f$(a)=h$: print chr$(142)
540 return
550 cc$=right$(g$, 24): id$=mid$(cc$, 20,
    5): print "[6xSPACE], [CTRL 9]"; cc$: r
    eturn
560 print
570 print "[5xSPACE]"; g$; "blokken [SPACE]
    ] vrij." : cc$=cc$+g$+" [SPACE] blokken
    [SPACE] vrij. [2xSPACE]" + da$: return
580 : return
590 rem *****
592 rem * 1 kolom uitprinten *
596 rem *****
600 open 1, 4
610 print #1: cmd 1: print chr$(27) chr$(15)
    ;: print chr$(27) "s" chr$(0);
620 print #1, chr$(27) "a" chr$(5); " [17xSP

```

```

ACE] naam [SPACE] : [SPACE]"; mid$(cc$,
2, 13);
630 print #1, "[SPACE] nr.:"; id$
640 print #1, "[17xSPACE]=====
    "
650 print #1, "[17xSPACE]" mid$(cc$, 28, 16
    )
660 print #1, "[17xSPACE]=====
    "
670 a=a+1
690 for x=1 to int(a/1)
700 print #1, "[20xSPACE]" f$(x); " [3xSPAC
    E]";
710 print #1, f$(x+int(a/1)); " [3xSPACE]"
    ;
720 print #1, f$(+2*int(a/1))
730 next: goto 750
740 print #1, mid$(cc$, 28, 16)
750 close 1
760 print chr$(30) chr$(14) chr$(147): run
    10
770 return
780 rem *****
790 rem * 2 kolommen uitprinten *
800 rem *****
810 open 1, 4
820 print #1: cmd 1: print chr$(27) chr$(15)
    ;: print chr$(27) "s" chr$(0);
830 print #1: print #1
840 print #1, chr$(27) "a" chr$(5); " [7xSPA
    CE] naam [SPACE] : [SPACE]"; mid$(cc$, 2
    , 16);
850 print #1, "[SPACE] nr. [SPACE] : [SPACE]
    "; id$;
860 print #1, "[4xSPACE]" mid$(cc$, 28, 16)
870 print #1, "[7xSPACE]=====
    =====
    ====="
890 ifa/2<>int(a/2) and a-2*int(a/2)=1 then
    ena=a+2: goto 900
900 for x=1 to int(a/2)
910 print #1, "[8xSPACE]" f$(x); " [3xSPACE]
    ";
920 print #1, f$(x+int(a/2)); " [3xSPACE]"
    ;
930 print #1, f$(x+2*int(a/2))
940 next: goto 960
950 print #1, mid$(cc$, 28, 16)
960 close 1
970 print chr$(30) chr$(14) chr$(147): run
    10
980 return
990 :
1000 rem *****
1010 rem * 3 kolommen uitprinten *
1020 rem *****
1030 open 1, 4
1040 print #1: cmd 1: print chr$(27) chr$(15)
    ;: print chr$(27) "s" chr$(0);
1050 print #1, chr$(27) "a" chr$(5); " [2xSPA
    CE] naam [SPACE] : [SPACE]"; mid$(cc$, 2
    , 16);
1060 print #1, "[5xSPACE] nr. [SPACE] : [SPAC
    E]"; id$;
1070 print #1, "[6xSPACE]" mid$(cc$, 28, 16)
1080 print #1, "[SPACE]=====
    =====
    =====";
1090 print #1, "=====
    -----"
1100 ifa/3<>int(a/3) and a-3*int(a/3)=2 then
    ena=a+3: goto 1120
1110 ifa/3<>int(a/3) and a-3*int(a/3)=1 then
    ena=a+3: goto 1120
1120 for x=1 to int(a/3)

```



```

1130 print#1,"[SPACE]"f$(x);"[3xSPACE]"
;
1140 print#1,f$(x+int(a/3));"[3xSPACE]"
;
1150 print#1,f$(x+2*int(a/3))
1160 next:goto1180
1170 print#1,mid$(cc$,28,16)
1180 close1
1190 printchr$(30)chr$(14)chr$(147):run
10
1200 return

```

** EINDE LISTING files uitprinten **

```

REGEL 1050 74
REGEL 1060 104
REGEL 1070 39
REGEL 1080 228
REGEL 1090 95
REGEL 1100 251
REGEL 1110 250
REGEL 1120 135
REGEL 1130 6
REGEL 1140 147
REGEL 1150 183
REGEL 1160 15
REGEL 1170 227
REGEL 1180 209
REGEL 1190 106
REGEL 1200 142

```

REGEL 1 85	REGEL 490 54
REGEL 2 255	REGEL 500 254
REGEL 3 156	REGEL 510 15
REGEL 4 49	REGEL 520 13
REGEL 5 167	REGEL 530 5
REGEL 6 162	REGEL 540 164
REGEL 7 85	REGEL 550 142
REGEL 10 166	REGEL 560 198
REGEL 20 108	REGEL 570 153
REGEL 30 51	REGEL 580 62
REGEL 40 63	REGEL 590 200
REGEL 50 85	REGEL 592 43
REGEL 60 18	REGEL 594 168
REGEL 70 96	REGEL 596 43
REGEL 80 153	REGEL 600 48
REGEL 90 210	REGEL 610 76
REGEL 100 108	REGEL 620 71
REGEL 110 181	REGEL 630 45
REGEL 120 153	REGEL 640 95
REGEL 130 184	REGEL 650 39
REGEL 140 96	REGEL 660 95
REGEL 150 206	REGEL 670 15
REGEL 160 120	REGEL 690 133
REGEL 170 123	REGEL 700 6
REGEL 180 165	REGEL 710 145
REGEL 185 145	REGEL 720 93
REGEL 190 31	REGEL 730 225
REGEL 200 43	REGEL 740 227
REGEL 210 1	REGEL 750 209
REGEL 220 43	REGEL 760 106
REGEL 230 83	REGEL 770 142
REGEL 240 219	REGEL 780 169
REGEL 250 87	REGEL 790 137
REGEL 260 138	REGEL 800 169
REGEL 270 242	REGEL 810 48
REGEL 275 16	REGEL 820 76
REGEL 280 132	REGEL 830 204
REGEL 290 58	REGEL 840 74
REGEL 300 207	REGEL 850 104
REGEL 310 121	REGEL 860 39
REGEL 320 207	REGEL 870 84
REGEL 330 217	REGEL 890 202
REGEL 340 218	REGEL 900 134
REGEL 350 120	REGEL 910 6
REGEL 360 44	REGEL 920 146
REGEL 370 186	REGEL 930 182
REGEL 380 114	REGEL 940 228
REGEL 390 8	REGEL 950 227
REGEL 400 32	REGEL 960 209
REGEL 410 108	REGEL 970 106
REGEL 420 192	REGEL 980 142
REGEL 430 54	REGEL 990 58
REGEL 440 156	REGEL 1000 169
REGEL 450 54	REGEL 1010 138
REGEL 460 32	REGEL 1020 169
REGEL 470 58	REGEL 1030 48
REGEL 480 46	REGEL 1040 76

nabestellen OUDE NUMMERS

Hieronder volgt een overzicht van de
verschenen en nog beschikbare nummers.

Jaar	Aantal verschenen nummers	Nog beschikbaar
1984	5	niets
1985	10	alle nummers
1986	9	alle nummers

Reeds verschenen nummers zijn na te bestellen.
Maar, helaas, NIET per brief, kaart of telefonisch.
We leveren die oude nummers alleen bij
voorstelbetaling op onze giro 1585491 t.n.v. Sala
Communications/SAC te Blaricum of op onze
bank in België BBL nr. 310050602562.

De prijs is f 6,75 per nummer
(dus aangeven welk nummer).

Doe iets leuks met uw computer-hobby en zendt uw zelfgemaakte programma in.

PRIJSVRAAG voorwaarden:

- ° maximale lengte van het programma 150 regels.
Inzendingen voor serieuze toepassingen mogen wat groter van lengte zijn.
- ° Alleen originele programma's mogen worden ingezonden, de inzender blijft hiervoor verantwoordelijk! Inzending alleen op een magnetisch medium (cassette of diskette).
- ° Geen inzendingen die u ook naar andere bladen stuurt of heeft gestuurd.

Zo kunt u winnen:

- 1) Als beste inzender in uw categorie.
- 2) Wanneer we uw programma goed genoeg vinden om te publiceren. In dat geval krijgt u daarvoor een vergoeding uitgekeerd.

Wat mag het zijn?

Utilities, spelletjes, doe-programma's en programma's voor serieuze toepassingen.

- ° Iedere 10e inzender ontvangt een verrassing!!

Wat is er te winnen:

- ° Een kleuren monitor.
- ° Een monochrome monitor..
- ° Een diskdrive
- ° Een groot aantal prachtige software pakketten.

Als extra hebben we een categorie voor de *mooiste, beste en voor een groot publiek interessante hardware-aanpassing*. Het moet voor iedereen die geen twee linkerhanden heeft na te maken zijn. Inzendingen moeten naast een goed bouwschema ook een materiaalijst bevatten. Het ontwerp bijsluiten hoeft niet (mag wel) als het nodig is nemen wij daar contact met u over op.

N.B. Door inzending stemt men toe in publicatie, ook in elektronische vorm. De vergoeding bij plaatsing wordt door de redactie bepaald. Inzenders krijgen na ontvangst van hun inzending van ons een andere cassette of diskette toegestuurd, dus hou zelf een kopie van het programma.
Inzendingen: PB 112, 1260 AC Blaricum o.v.v. **"PRIJSVRAAG Commodore"**.

Missers

P.J. Bokma uit Heerenveen maakte ons attent op de volgende misser, en verstrekke een tip betreffende het Amiga-programma **Dataspin** in Commodore Info van oktober/november 1987. In de toelichting (dus niet in het programma) staat dat om te eindigen een " - " moet worden ingetoetst. Dit is niet juist. Het moet zijn: " . " (een punt).

Diegenen die wat in Amiga-Basic thuis zijn zullen direkt gezien hebben dat achter het label "verder4" een dubbele punt geplaatst moet worden: **"verder4:"**

Nog een tip: met behulp van de Absoft A.C. Basic compiler kan het programma als volgt worden aangepast:

Ten eerste moet dan als eerste regel een label worden geplaatst, bv. "start:". Vervolgens kunnen het aantal records dat onder het label "dimensionering:" staat worden aangepast: `if vld=2 then a1=2001, if vld=3 then a1=1601 etc.` Dit kan omdat de compiler meer geheugen beschikbaar stelt aan het programma. Wanneer men na het compileren vanuit het hoofdmenu "6" kiest (print vrij geheugen) dan blijkt dit meer dan 200 KB te zijn.

Sluitingsdatum inzendingen: 31 december 1987

POWERBOOT

Dit programma werkt alleen in combinatie met de powercardridge. Na het runnen verschijnt de dyrectory op het scherm met voor de titel een cijfer. Om een programma te willen laten opstarten is het voldoende om het corresponderende cijfer in te toetsen. Het programma is alleen geschikt voor diskettes met niet te veel programma's op de schijf, maar is voor veel schijven goed te gebruiken. Het is een creatie van R.A. Wolswijk uit Zeist.

```

1 rem powerboot / commodore 64
2 rem door r.a. wolswijk
3 rem uit zeist / 03404-23295
4 rem
5 pufef:rem ** boot voor powercart.
**
6 print"[SHIFT CLR]";:catalog:b=1069
:print"[HOME]";:ifpeek(1024)<>48the
nb=1029:print"[HOME]";
7 dimd(50):t=40:x=1029:c=b-t
8 a=a+1:b=b+t:printstr$(a);":ifpee
k(b)=34then8

```

```

9 print"[HOME]";tab(28)"[CTRL 9]ISPA
CELwelk[SPACE]prgnr.[CTRL 0]"
10 printtab(29);:inputp$:ifp$="↑"then
run6
11 p=val(p$):ifp<10rp>athen:pufef:pri
nt"[SHIFT CLR]";:new
12 c=c+p*t:fork=1to17:d(k)=peek(c+k):
ifpeek(c+k)=34then14
13 next
14 print"[SHIFT CLR]load";chr$(34):fo
r1=1to17:ifd(1)=0then16
15 pokex,d(1):x=x+1:next
16 pokex,44:pokex+1,56:poke631,19:pok
e632,13:poke633,13
17 poke634,82:poke635,85:poke636,78:p
oke637,13:poke198,7:pufef:end
** EINDE LISTING powerboot

```

REGEL 1	142	REGEL 10	50
REGEL 2	44	REGEL 11	195
REGEL 3	108	REGEL 12	115
REGEL 4	143	REGEL 13	130
REGEL 5	181	REGEL 14	214
REGEL 6	31	REGEL 15	47
REGEL 7	164	REGEL 16	24
REGEL 8	96	REGEL 17	141
REGEL 9	51		

Springbeest

Springbeest is een programma gemaakt door T. Stegman uit Zeist. Je bestuurt een kikker die op het ijs heen en weer springt. Er zijn boven het ijs vliegen die de kikker moet proberen te vangen. Iedere keer dat de kikker na een sprong op het ijs neer valt verschijnt er op die plaats een wak als hij weer op die plaats neer komt is het spel afgelopen. Als alle vliegen zijn opgegeten ga je door naar een volgend level

```

5      ho=0
10     forq=0to62:readj:poke832+q,j:next
20     v=53248:e=6:f=63
30     poke53280,14:poke53281,0
40     pokev+21,1:poke2040,13:pokev+39,5
50     s=54272:pokes+24,15
60     forq=stos+23:pokeq,0:next
70     pokes+2,100:pokes+5,9
80     pokes,37:pokes+6,48
90     pokes+4,17:pokes+1,55
100    print"[SHIFT CLR][22xCRSR-DOWN][SPACE]
      [CTRL 9][38xCOM I][CTRL 0][SPACE]"
110    forj=1toe:q=int(rnd(0)*40)+1904:if
      peek(q)=32thenj=j-1
120    pokeq,32:next
130    forj=1tof:q=1024+int(rnd(0)*480)
140    ifpeek(q)=81thenj=j-1
150    pokeq,81:pokeq+54272,int(rnd(0)*15
      -1):next
160    x=170:y=102:ti$="000000"
170    gosub280:pokes+4,17
180    gosub580:gosub190:goto170
190    j=peek(56320)
200    h=sgn(jand4)-sgn(jand8)
210    x=x-h*8:pokev,xand255
220    ifx<30thenx=30
230    ifx>310thenx=310
240    y=y-10:pokev+16,int(x/256):pokes+1
      ,y
250    pokev+1,y:ifpeek(v+31)=landy<190th
      engosub490:return
260    ify<>52then190
270    return
280    j=peek(56320)
290    h=sgn(jand4)-sgn(jand8)
300    y=y+10:x=x+h*8:pokev,xand255
310    ifx<30thenx=30
320    ifx>310thenx=310
330    pokev+16,int(x/256):pokes+1,y
340    ifpeek(v+31)=landy>190theny=222:po
      kev+1,y:return
350    ifpeek(v+31)=1thengosub490
360    pokev+1,y:ify<>232then280
370    forl=xtos+367step3:pokev+16,int(l/25
      6):pokev,land255:pokes+1,land37+k
380    next:pokes+6,0:o=o+k:t=int(ti/60):
      poke53272,21
390    bs=ti$:print"[SHIFT CLR][3xCRSR-DO
      WN]je[SPACE]hebt[SPACE]net"k"vlieg
      en[SPACE]ingeslikt[SPACE]in:
400    print"[6xSPACE]":print t"seconden"
401    k=0:t=0
410    print"[4xCRSR-DOWN]je[SPACE]score[
      SPACE]is:"o
412    if o>ho then ho=o:o=0
413    print"[3xCRSR-DOWN]de[SPACE]hoogst
      e[SPACE]score[SPACE]van[SPACE]de[S
      PACE]dag[SPACE]is"ho
420    print"[8xCRSR-DOWN]wil[SPACE]je[SP
      ACE]nog[SPACE]een[SPACE]keer[SPACE]
      spelen";:inputq$

```

```

430    if q$="j"then goto100
440    if q$="n"then end
450    ifk<>fthenend
460    e=e+4:k=0:f=f+4:print"press[SPACE]
      any[SPACE]key!"
470    geta$:ifa$=""then470
480    ti$=b$:goto50
490    qq=int(((x-24)/8)+int((y-53)/8)*40
      )+984
500    forj=0to4:forq=0to2
510    d=qq+j+q*40
520    ifpeek(d)=32then550
530    ifpeek(d)=81thenk=k+1
540    pokes+4,129:pokes+4,128:poked,32
550    nextq,j:pokes+4,17
560    cc=0:ify<190thencc=1
570    return
580    qq=int(((x-24)/8)+int((y-53)/8)*40
      )+1064
590    forj=0to3
600    d=qq+j
610    ifpeek(d)=32then630
620    pokes+4,129:pokes+4,128:poked,32
630    nextj:pokes+4,17:return
640    data7,0,112,5,127,80,5
650    data255,208,15,190,248,95,255
660    data253,111,0,123,127,255,255
670    data111,255,251,55,255,246,59
680    data255,238,28,0,28,6,193
690    data176,2,193,160,126,193,191
700    data0,193,128,17,128,196,127
710    data0,127,0,0,0,0,0
720    data0,0,0,0,0,0,0

```

** EINDE LISTING springbeest **

EGEL	5	121	EGEL	380	186
EGEL	10	102	EGEL	390	65
EGEL	20	16	EGEL	400	151
EGEL	30	90	EGEL	401	157
EGEL	40	28	EGEL	410	81
EGEL	50	207	EGEL	412	204
EGEL	60	23	EGEL	413	155
EGEL	70	235	EGEL	420	249
EGEL	80	28	EGEL	430	157
EGEL	90	241	EGEL	440	107
EGEL	100	150	EGEL	450	167
EGEL	110	167	EGEL	460	109
EGEL	120	53	EGEL	470	104
EGEL	130	208	EGEL	480	1
EGEL	140	211	EGEL	490	251
EGEL	150	100	EGEL	500	73
EGEL	160	139	EGEL	510	151
EGEL	170	189	EGEL	520	58
EGEL	180	230	EGEL	530	199
EGEL	190	15	EGEL	540	255
EGEL	200	13	EGEL	550	223
EGEL	210	46	EGEL	560	227
EGEL	220	13	EGEL	570	142
EGEL	230	109	EGEL	580	33
EGEL	240	0	EGEL	590	132
EGEL	250	90	EGEL	600	140
EGEL	260	240	EGEL	610	57
EGEL	270	142	EGEL	620	255
EGEL	280	15	EGEL	630	42
EGEL	290	13	EGEL	640	242
EGEL	300	215	EGEL	650	105
EGEL	310	13	EGEL	660	80
EGEL	320	109	EGEL	670	98
EGEL	330	86	EGEL	680	155
EGEL	340	207	EGEL	690	96
EGEL	350	182	EGEL	700	48
EGEL	360	167	EGEL	710	69
EGEL	370	102	EGEL	720	219

Vaste Hand C128

De Heer Y. Rozijn uit Amsterdam heeft voor ons een programma gemaakt waarbij je je zenuwen goed in bedwang moet hebben om het spel tot een goed einde te brengen. De opdracht is om de ring zo snel mogelijk van links naar rechts te sturen. De toevallige bewegingen van de ring nemen toe naarmate de tijd verstrijkt. Snelheid en een vaste hand zijn dus geboden.

```

1      rem *****
2      rem *       vaste hand       *
3      rem *       commodore 128     *
4      rem * door y. rozijn, amsterdam *
5      rem *****
10     gosub 900:n=1:t=-1
20     do:color 5,8
30     : movespr 1,0,79:sprite 1,1,2:move
      spr 1,90#1
40     : do:j=joy(2):m=n
50     : : if j=1 then m=n-1:if m<1 then
      m=6
60     : : if j=5 then m=n+1:if m>6 then
      m=1
70     : : if m<>n then char 1,1,18+n,"[2
      xSPACE]":char 1,1,18+m,"->":n=m
80     : : if j=3 or j=7 then begin
90     : : : t=not t:if t then char 1,1,1
      2,"[SPACE]wel":else char 1,1,12,"g
      een"
100    : : bend
110    : : if (jand15)>0 then do:loop unt
      il joy(2)=0
120    : loop until j and 128
130    : gosub 660:s=s/10:color 5,11:char
      1,30,19
140    : print using "####.#";s:char 1,30
      ,21,str$(n)
150    : if x then char 1,25,23,"niet[SPA
      CE]voltooid":else begin
160    : : char 1,25,23,chr$(27)+"q":colo
      r 5,8
170    : : if s<hs(n) then hs(n)=s:char 1
      ,14,18+n:print using "####.#";hs(n
      );
180    : bend
190    loop
200    end
210    rem **** logo ****
220    char 1,6,0,"[COM A][25xSHIFT *][CO
      M S]"
230    char 1,6,1,"[SHIFT -][2xSPACE]v[SP
      ACE]a[SPACE]s[SPACE]t[SPACE]e[5xSP
      ACE]h[SPACE]a[SPACE]n[SPACE]d[2xSP
      ACE][SHIFT -]"
240    char 1,6,2,"[COM Z][25xSHIFT *][CO
      M X]"
250    return
260    rem **** baan 1 ****
270    draw 1,6,110 to 313,110:y=110
280    return
290    rem **** baan 2 ****
300    draw 1,6,170 to 30,170
310    circle 1,160,170,130,120,270,450,,
      45
320    draw 1,290,170 to 313,170:y=170
330    return
340    rem **** baan 3 ****
350    draw 1,6,110 to 30,110
360    circle 1,90,110,60,60,270,450
370    draw 1,150,110 to 170,110
380    circle 1,230,110,60,60,90,270

```

```

390    draw 1,290,110 to 313,110
400    y=110
410    return
420    rem **** baan 4 ****
430    draw 1,6,170 to 30,170 to 30,50 to
      82,50 to 82,170 to 132,170
440    drawto 132,50 to 184,50 to 184,170
      to 238,170 to 238,50 to 290,50
450    drawto 290,170 to 313,170:y=170
460    return
470    rem **** baan 5 ****
480    draw 1,6,170 to 140,170 to 140,150
490    circle 1,140,100,110,50,180,360
500    draw 1,140,50 to 180,50
510    circle 1,180,100,110,50,0,180
520    draw 1,180,150 to 180,170 to 313,1
      70
530    y=170
540    return
550    rem **** baan 6 ****
560    draw 1,6,110 to 30,110
570    circle 1,160,110,130,60,270,360
580    circle 1,160,110,100,60,0,90
590    circle 1,210,110,50,30,90,270
600    circle 1,160,110,130,60,90,180
610    circle 1,160,110,100,60,180,270
620    circle 1,110,110,50,30,270,450
630    draw 1,290,110 to 313,110
640    y=110
650    return
660    rem **** spelronde ****
670    if t then def fnr(x)=rnd(1)*(2*x+1
      )-x:else def fnr(x)=0
680    sprite 1,0:movespr 1,0#0:graphic 1
      ,1:color 1,2:gosub 210:color 1,8
690    on n gosub 260,290,340,420,470,550
700    color 1,16:char 1,2,24,"baan"+str$(
      n):char 1,15,24,"tijd[7xSPACE](ti
      ende[SPACE]sec)"
710    movespr 1,24,y+44:sprite 1,1,2:p=0
      :tis="000000"
720    do
730    : s=int(ti/6):char 1,19,24,str$(s)
740    : j=joy(2)and 15:p=p-.01*(p<2)
750    : y=dy(j)+fnr(p):z=dz(j)+fnr(p)
760    : x=bump(2)=0:if x=0 then movespr
      1,+y,+z
770    loop until x or rsppos(1,0)>329
780    if x then begin
790    : sound 1,10000,120,2,8000,1000,1
800    : c=11:for i=1 to 49:sprite 1,1,c:
      c=13-c:for j=1to15:next:next
810    bend:else begin
820    : sprite 1,1,14
830    : sound 1,6000,52:for i=1 to 200:n
      ext
840    : sound 2,7559,86:for i=1 to 200:n
      ext
850    : sound 3,8989,68:for i=1 to 200:n
      ext
860    : sound 1,12000,50
870    bend
880    sleep 2:sprite 1,0:graphic 0
890    return
900    rem **** init ****
910    color 0,1:color 4,1:color 5,2:grap
      hic 1,1:graphic 0,1
920    gosub 210:print:print:color 5,16
930    print "beweeg[SPACE]de[SPACE]ring[
      SPACE]zo[SPACE]snel[SPACE]mogelijk
      [SPACE]van"
940    print "links[SPACE]naar[SPACE]rech
      ts[SPACE]maar[SPACE]verlaat[SPACE]
      de[SPACE]lijn"

```

```

950 print "niet.":print
960 print "de[SPACE]toevallige[SPACE]b
    ewegingen[SPACE]van[SPACE]de[SPACE]
    ring"
970 print "nemen[SPACE]toe[SPACE]naarm
    ate[SPACE]de[SPACE]tijd[SPACE]vers
    trijkt."
980 print "een[SPACE]vaste[SPACE]hand[
    SPACE]is[SPACE]dus[SPACE]geboden!"
990 color 5,4:with 2:vol 15
1000 circle 1,6,6,6,6:shape a$.0,0,23,2
    1:sprsave a$,1
1010 sprite 1,0,2,1,0,0,0
1020 for i=1 to 8:read dy(i),dz(i):next
1030 data 0,-3,3,-3,3,0,3,3,0,3,-3,3,-3
    ,0,-3,-3
1040 for i=1 to 6:hs(i)=99999:next
1050 print:print "[6xSPACE]toevallige[S
    PACE]bewegingen[SPACE] (joyst[SPACE]

```

```

]l/r)"
1060 print:print "[2xSPACE]selekteer[SP
    ACE]een[SPACE]baan[SPACE]met[SPACE]
    de[SPACE]joystick"
1070 print "[2xSPACE]druk[SPACE]op[SPAC
    E]de[SPACE]vuurknop[SPACE]om[SPACE]
    te[SPACE]beginnen"
1080 color 5,8:char 1,2,12,"wel":char 1
    ,6,17,"kortste[SPACE]tijden":char
    1,1,19,"->"
1090 for i=1 to 6:char 1,5,18+i,"baan"+
    str$(i):next
1100 color 5,11:char 1,25,18,"laatste"
1110 char 1,25,19,"tijd":char 1,25,21,
    "(baan[2xSPACE])"
1120 return

```

** EINDE LISTING vaste hand 128 **

```

REGEL 1 81
REGEL 2 129
REGEL 3 35
REGEL 4 68
REGEL 5 81
REGEL 10 173
REGEL 20 165
REGEL 30 234
REGEL 40 52
REGEL 50 206
REGEL 60 207
REGEL 70 166
REGEL 80 206
REGEL 90 96
REGEL 100 139
REGEL 110 120
REGEL 120 182
REGEL 130 87
REGEL 140 117
REGEL 150 46
REGEL 160 79
REGEL 170 45
REGEL 180 81
REGEL 190 236
REGEL 200 128
REGEL 210 16
REGEL 220 93
REGEL 230 152
REGEL 240 107
REGEL 250 142

```

```

REGEL 260 34
REGEL 270 6
REGEL 280 142
REGEL 290 35
REGEL 300 7
REGEL 310 100
REGEL 320 125
REGEL 330 142
REGEL 340 36
REGEL 350 251
REGEL 360 20
REGEL 370 144
REGEL 380 16
REGEL 390 148
REGEL 400 157
REGEL 410 142
REGEL 420 37
REGEL 430 14
REGEL 440 46
REGEL 450 196
REGEL 460 142
REGEL 470 38
REGEL 480 52
REGEL 490 106
REGEL 500 54
REGEL 510 5
REGEL 520 157
REGEL 530 163
REGEL 540 142
REGEL 550 39

```

```

REGEL 560 251
REGEL 570 112
REGEL 580 212
REGEL 590 10
REGEL 600 64
REGEL 610 109
REGEL 620 57
REGEL 630 148
REGEL 640 157
REGEL 650 142
REGEL 660 139
REGEL 670 227
REGEL 680 22
REGEL 690 221
REGEL 700 100
REGEL 710 6
REGEL 720 235
REGEL 730 204
REGEL 740 149
REGEL 750 128
REGEL 760 180
REGEL 770 240
REGEL 780 160
REGEL 790 189
REGEL 800 176
REGEL 810 60
REGEL 820 94
REGEL 830 163
REGEL 840 191
REGEL 850 200

```

```

REGEL 860 245
REGEL 870 23
REGEL 880 79
REGEL 890 142
REGEL 900 19
REGEL 910 37
REGEL 920 175
REGEL 930 87
REGEL 940 47
REGEL 950 14
REGEL 960 203
REGEL 970 123
REGEL 980 240
REGEL 990 125
REGEL 1000 6
REGEL 1010 97
REGEL 1020 161
REGEL 1030 73
REGEL 1040 129
REGEL 1050 46
REGEL 1060 29
REGEL 1070 64
REGEL 1080 196
REGEL 1090 1
REGEL 1100 155
REGEL 1110 79
REGEL 1120 142

```

mastermind C 128

Tijn Kemps uit Schiedam gebruikte het bekende geheugenspel mastermind als uitgangspunt voor een computer versie. Een uitleg is overbodig. De enige opdracht is om in zo min mogelijk beurten de kleurcode, die de computer heeft gekozen te raden. Het programma is met de checksum te kontroleren maar alleen in de 64 mode.

```

10 rem *****
11 rem *
12 rem * mastermind cbm-128 *
13 rem *
14 rem * door tijn kemps *
15 rem *
22 rem *****
23 :
100 printchr$(142):graphic0,1:fast
101 poke53280,0:poke53281,0
102 print"[3xCRSR-DOWN][SHIFT CLR]dit[

```

```

SPACE]programma[SPACE]is[SPACE]ges
chreven[SPACE]door[SPACE]:"
103 print "[2xCRSR-DOWN][CTRL 9][CTRL
    2]tijn[SPACE]kemps[CTRL 0][SPACE]u
    it[SPACE]schiedam"
104 slow
105 sleep5:print"[SHIFT CLR]":goto206
106 color0,1:fast:graphic1,1:color0,1
107 color1,8
108 char1,13,0,"*[SPACE]master-mind[SP
    ACE]*"
109 color1,11
110 char1,3,2,"UCCCCCCI"
111 char1,3,3,"B*[SPACE]*[SPACE]*[SPAC
    E]*B"
112 char1,3,4,"JCCCCCCK"
113 char1,0,6,"01[2xSPACE]W[SPACE]W[SP
    ACE]W[SPACE]W[2xSPACE].[SPACE]."
114 char1,0,8,"02[2xSPACE]W[SPACE]W[SP
    ACE]W[SPACE]W[2xSPACE].[SPACE]."
115 char1,0,10,"03[2xSPACE]W[SPACE]W[S
    PACE]W[SPACE]W[2xSPACE].[SPACE]."

```

```

116 char1,0,12,"04[2xSPACE]W[SPACE]W[S
    PACE]W[SPACE]W[2xSPACE].[SPACE]."
117 char1,0,14,"05[2xSPACE]W[SPACE]W[S
    PACE]W[SPACE]W[2xSPACE].[SPACE]."
118 char1,0,16,"06[2xSPACE]W[SPACE]W[S
    PACE]W[SPACE]W[2xSPACE].[SPACE]."
119 char1,0,18,"07[2xSPACE]W[SPACE]W[S
    PACE]W[SPACE]W[2xSPACE].[SPACE]."
120 char1,0,20,"08[2xSPACE]W[SPACE]W[S
    PACE]W[SPACE]W[2xSPACE].[SPACE]."
121 char1,0,22,"09[2xSPACE]W[SPACE]W[S
    PACE]W[SPACE]W[2xSPACE].[SPACE]."
122 char1,0,24,"10[2xSPACE]W[SPACE]W[S
    PACE]W[SPACE]W[2xSPACE].[SPACE]."
123 color1,2:char1,22,6,"Q":char1,22,8
    ,"1"
124 color1,3:char1,24,6,"Q":char1,24,8
    ,"2"
125 color1,6:char1,26,6,"Q":char1,26,8
    ,"3"
126 color1,7:char1,28,6,"Q":char1,28,8
    ,"4"
127 color1,8:char1,30,6,"Q":char1,30,8
    ,"5"
128 color1,5:char1,32,6,"Q":char1,32,8
    ,"6"
129 slow:cd$=""
130 forcd=1to4
131 c(cd)=int(6*rnd(1)+1)
132 nextcd
133 fora=1to3:forb=a+1to4
134 ifc(a)=c(b)then129
135 nextb,a
136 z$(1)="01":z$(2)="02":z$(3)="03":z
    $(4)="04":z$(5)="05":z$(6)="06":z$
    (7)="07":z$(8)="08":z$(9)="09":z$(
    10)="10"
137 forbe=1to10
138 color1,4:char1,20,10,z$(be)+"[10xS
    PACE]"
139 be$=""
140 forkk=1to4
141 getff$:ffff$<"1"orfff$>"6"then141
142 be$=be$+fff$:tb$(kk)=fff$
143 forcc=kk-1to1step-1
144 ifff$=tb$(cc)andcc<>0then138
145 nextcc
146 ifff$="1"thencolor1,2
147 ifff$="2"thencolor1,3
148 ifff$="3"thencolor1,6
149 ifff$="4"thencolor1,7
150 ifff$="5"thencolor1,8
151 ifff$="6"thencolor1,5
152 char1,22+2*kk,10,"Q"
153 nextkk
154 color1,15:char1,20,12,"zeker[SPACE
    ]weten[SPACE]?[SPACE](j/n)"
155 geta$:ifa$<>"j"anda$<>"n"then155
156 char1,20,12,"[19xSPACE]"
157 ifa$="n"then138
158 forcc=1to4
159 iftb$(cc)="1"thencolor1,2
160 iftb$(cc)="2"thencolor1,3
161 iftb$(cc)="3"thencolor1,6
162 iftb$(cc)="4"thencolor1,7
163 iftb$(cc)="5"thencolor1,8
164 iftb$(cc)="6"thencolor1,5
165 char1,2+cc*2,4+be*2,"Q"
166 nextcc
167 g=0:h=0
168 forgg=1to4
169 ifval(tb$(gg))=c(gg)theng=g+1
170 nextgg
171 forgg=1to4:forhg=1to4
172 ifval(tb$(gg))=c(hg)andgg<>hgthenh
    -h+1
173 nexthg,gg
174 color1,14:char1,12,4+be*2,str$(g)
175 color1,11:char1,14,4+be*2,str$(h)
176 ifg=4then190
177 nextbe
178 forcc=1to4
179 ifc(cc)=1thencolor1,2
180 ifc(cc)=2thencolor1,3
181 ifc(cc)=3thencolor1,6
182 ifc(cc)=4thencolor1,7
183 ifc(cc)=5thencolor1,8
184 ifc(cc)=6thencolor1,5
185 char1,2+cc*2,3,"Q"
186 nextcc
187 color1,4:char1,20,6,"helaas[SPACE]
    !![13xSPACE]"
188 char1,20,8,"u[SPACE]heeft[SPACE]ve
    rloren."
189 goto202
190 forcc=1to4
191 iftb$(cc)="1"thencolor1,2
192 iftb$(cc)="2"thencolor1,3
193 iftb$(cc)="3"thencolor1,6
194 iftb$(cc)="4"thencolor1,7
195 iftb$(cc)="5"thencolor1,8
196 iftb$(cc)="6"thencolor1,5
197 char1,2+cc*2,3,"Q"
198 nextcc
199 sleep3
200 color1,4:char1,20,6,"gefeliciteerd
    [SPACE]!!"
201 char1,20,8,"u[SPACE]hebt[SPACE]gew
    onnen[SPACE]!!"
202 char1,20,10,"spatie[SPACE]voor[SPA
    CE]opnieuw."
203 char1,20,12,"return[SPACE]voor[SPA
    CE]uitleg."
204 geta$:ifa$<>"[SPACE]"anda$<>chr$(1
    3)then204
205 ifa$="[SPACE]"then106
206 graphic0,1:print"[SHIFT CLR]u[SPAC
    E]moet[SPACE]de[SPACE]juiste[SPACE
    ]kleurcode[SPACE]raden[SPACE]in[SP
    ACE]zo[2xSPACE]min[SPACE]mogenlijk
    [SPACE]beurten."
207 print"[CRSR-DOWN]het[SPACE]eerste[
    SPACE]getal[SPACE]achter[SPACE]uw[
    SPACE]beurt[SPACE]geeft[2xSPACE]de
    [SPACE]juiste[SPACE]kleuren[SPACE]
    op[SPACE]de[SPACE]juiste[SPACE]pla
    ats[3xSPACE]aan[SPACE]en[SPACE]het
    [SPACE]tweede[SPACE]getal[SPACE]de
    [SPACE]juiste"
208 print"kleuren[SPACE]die[SPACE]niet
    [SPACE]op[SPACE]de[SPACE]juiste[SP
    ACE]plaats[SPACE]zitten."
209 print"[CRSR-DOWN]u[SPACE]moet[SPAC
    E]steeds[SPACE]vier[SPACE]verschil
    lende[SPACE]kleurengebruiken[SPACE
    ]!"
210 print"[CRSR-DOWN]veel[SPACE]plezie
    r[SPACE]!!"
211 print"[CRSR-DOWN]druk[SPACE]spatie
    [SPACE]voor[SPACE]spelen."
212 geta$:ifa$<>"[SPACE]"then212
213 goto106

```

** EINDE LISTING mastermind 128 **

REGEL	10	249	REGEL	123	212	REGEL	154	221	REGEL	185	29
REGEL	11	227	REGEL	124	218	REGEL	155	110	REGEL	186	8
REGEL	12	113	REGEL	125	226	REGEL	156	158	REGEL	187	19
REGEL	13	227	REGEL	126	232	REGEL	157	119	REGEL	188	207
REGEL	14	204	REGEL	127	220	REGEL	158	194	REGEL	189	29
REGEL	15	227	REGEL	128	222	REGEL	159	96	REGEL	190	194
REGEL	22	249	REGEL	129	255	REGEL	160	98	REGEL	191	96
REGEL	23	58	REGEL	130	195	REGEL	161	102	REGEL	192	98
REGEL	100	74	REGEL	131	205	REGEL	162	104	REGEL	193	102
REGEL	101	37	REGEL	132	9	REGEL	163	106	REGEL	194	104
REGEL	102	30	REGEL	133	31	REGEL	164	104	REGEL	195	106
REGEL	103	141	REGEL	134	43	REGEL	165	45	REGEL	196	104
REGEL	104	36	REGEL	135	49	REGEL	166	8	REGEL	197	29
REGEL	105	67	REGEL	136	232	REGEL	167	141	REGEL	198	8
REGEL	106	37	REGEL	137	240	REGEL	168	202	REGEL	199	60
REGEL	107	124	REGEL	138	78	REGEL	169	208	REGEL	200	17
REGEL	108	226	REGEL	139	161	REGEL	170	16	REGEL	201	142
REGEL	109	166	REGEL	140	210	REGEL	171	207	REGEL	202	253
REGEL	110	49	REGEL	141	182	REGEL	172	3	REGEL	203	188
REGEL	111	107	REGEL	142	159	REGEL	173	203	REGEL	204	9
REGEL	112	42	REGEL	143	133	REGEL	174	122	REGEL	205	36
REGEL	113	88	REGEL	144	138	REGEL	175	122	REGEL	206	11
REGEL	114	91	REGEL	145	8	REGEL	176	249	REGEL	207	84
REGEL	115	133	REGEL	146	127	REGEL	177	9	REGEL	208	194
REGEL	116	136	REGEL	147	129	REGEL	178	194	REGEL	209	17
REGEL	117	139	REGEL	148	133	REGEL	179	165	REGEL	210	119
REGEL	118	142	REGEL	149	135	REGEL	180	167	REGEL	211	37
REGEL	119	145	REGEL	150	137	REGEL	181	171	REGEL	212	20
REGEL	120	139	REGEL	151	135	REGEL	182	173	REGEL	213	32
REGEL	121	142	REGEL	152	141	REGEL	183	175			
REGEL	122	136	REGEL	153	24	REGEL	184	173			

MT base C16

Peter Boersma maakte een zeer uitgebreide database voor de C16. Het gehele geheugengebied van de c16 wordt hierbij benut. Het programma is geheel menu gestuurd zodat een uitleg overbodig is.

```

10 rem *****
20 rem *----- mt-base c16 -----*
30 rem * door: *
40 rem *--p e t e r b o e r s m a--*
50 rem *****
60 rem **** mt-base c-16/+4 ****
70 poke806,103
80 color0,1:color1,2:color4,1
90 clr:dim v$(9),g$(100,9)
100 scnclr:printchr$(14)chr$(8)
110 printtab(14)"MT-BASE[SPACE]C-16"
120 print"[SHIFT *]pb86[31xSHIFT *]sss
    [SHIFT *]"
130 rem g=aantal gegevens
140 rem a=nr gegeven
150 rem k=zoeken op v$(k)
160 rem v=aantal vraag-gegevens
170 rem ***** start *****
180 print"[3xCRSR-DOWN][SPACE][1][SPAC
    E]bestand[SPACE]laden"
190 print"[CRSR-DOWN][SPACE][2][SPACE]
    invoer[SPACE]nieuwe[SPACE]vraag-ge
    gevens"
200 print"[CRSR-DOWN][SPACE][3][SPACE]
    invoer[SPACE]gegevens"
210 print"[CRSR-DOWN][SPACE][4][SPACE]
    bestand[SPACE]doorzoeken"
220 print"[CRSR-DOWN][SPACE][5][SHIFT
    SPACE]bestand[SPACE]saven"
230 print"[CRSR-DOWN][SPACE][6][SPACE]
    einde"
240 print"[2xCRSR-DOWN][SPACE][-][2xCR
    SR-LEFT]";

```

```

250 get i$:if i$="" then 250
260 i=int(val(i$))
270 if i<1 or i>6 then 250
280 print i$
290 for l=1 to 200:next l:l=0
300 on i gosub 840,330,550,960,700,165
    0
310 goto 100
320 rem ----- invoer nwe vraag-gegeve
    ns
330 scnclr:g=0
340 if v=0then 370
350 for a=1 to v
360 v$(v)="":next a
370 print"[2xCRSR-DOWN]Hoeveel[SPACE]v
    ragen?(1-9)
380 getv$:ifv$="" then 380
390 if v$=chr$(27) then return
400 v=val(v$):if v<1 or v>9 then 380
410 for a=1 to v
420 print"Vraag[SPACE]nr. ";a
430 inputv$(a)
440 next a
450 print"[CRSR-DOWN]Nog[SPACE]wijzige
    n?(j/n)"
460 getw$:ifw$=""then460
470 if w$=chr$(27) then goto 500
480 if w$="j" or w$="J"then 330
490 if w$<>"n" and w$<>"N" then 460
500 x=0:for f=1 to v
510 h=len(v$(f)):if h>x then x=h
520 next f
530 return
540 rem ----- invoer gegevens -----
550 scnclr:if fre(0)<150org>99thenretu
    rn
560 print"[SHIFT CLR][2xCRSR-DOWN]":g=
    g+1:rem aantal gegevens
570 for a=1 to v
580 printv$(a)":":inputg$(g,a)
590 next a

```

```

600 print"[2xCRSR-DOWN][2xSPACE][1][SPACE]meer[SPACE]gegevens[SPACE]invo
    eren
610 print"[CRSR-DOWN][2xSPACE][2][SPACE]MENU"
620 print"[2xCRSR-DOWN][2xSPACE][-][2xCRSR-LEFT]";
630 getk$:ifk$="" then 630
640 if k$=chr$(27) then return
650 k=int(val(k$))
660 if k=1 then 550
670 if k<>2 then 630
680 return
690 rem ----- saven bestand -----
700 scnclr
710 print"[3xCRSR-DOWN]SAVEN[SPACE]BESTAND:zeker[SPACE]weten(j/n)"
720 getj$:ifj$="" then 720
730 if j$=chr$(27) then return
740 if j$="n" or j$="N" then return
750 if j$<>"j" and j$<>"J" then 720
760 input"Bestands-naam";b$
770 if b$="" then b$="BZN"
780 b=b+1:open b,1,1
790 print#b,b$:print#b,v:for a=1 to v:
    print#b,v$(a):next:print#b,g:ifg=0
    then goto 810
800 for a=1 to g:forc=1to v:print#b,g$(a,c):next:next
810 closeb
820 return
830 rem ----- bestand laden -----
840 scnclr:print"[CRSR-DOWN][SPACE]BESTAND[SPACE]LADEN:zeker[SPACE]weten
    (j/n)"
850 getkeya$:ifa$<>"j"anda$<>"J"thenreturn
860 for a=1 to g
870 for c=1 to v
880 g$(a,v)="" :v$(c)=""
890 next:next
900 b=b+1:open b,1,0
910 input#b,b$:input#b,v:for a=1 to v:
    input#b,v$(a):next:input#b,g:ifg=0
    then goto 930
920 for a=1 to g:forc=1to v:input#b,g$(a,c):next:next
930 closeb
940 return
950 rem ----- doorzoeken bestand -----
960 scnclr
970 print"[2xCRSR-DOWN][SPACE][0][SPACE]MENU"
980 print"[3xCRSR-DOWN][3xSPACE]Zoeken[SPACE]op:"
990 for i=1 to v
1000 print"[CRSR-DOWN][SPACE] [i] [2xSPACE]";v$(i)
1010 next i
1020 print"[2xCRSR-DOWN][SPACE][-][2xCRSR-LEFT]";
1030 getk$:ifk$="" then 1030
1040 if k$=chr$(27) then return
1050 if k$="0" then return
1060 k=int(val(k$))
1070 if k<1 or k>v then 1030
1080 printright$(str$(k),1)
1090 for l=1 to 200:next l
1100 scnclr
1110 print"[4xCRSR-DOWN]";v$(k);:input k$
1120 q=0:if k$="" then 1100
1130 for a=1 to g
1140 if g$(a,k)=k$ then gosub 1220
1150 if q=1 then return
1160 next a
1170 if q=1 then return
1180 print"[SHIFT CLR][2xCRSR-DOWN]Van[SPACE]"v$(k)": "k$
1190 print"niets[SPACE](meer)[SPACE]aanwezig"
1200 for l=1 to 2000:next:return
1210 rem --- printen gekozen gegevens ---
1220 scnclr:gosub 500
1230 for n=1 to v
1240 if len(g$(a,n))<39-x then 1260
1250 print v$(n)": :printtab(5)g$(a,n):
    goto 1270
1260 printv$(n);": :tab(x+1)g$(a,n)
1270 next n
1280 print"[CRSR-DOWN][SPACE]-s-[SPACE]schrappen"
1290 print"[SPACE]-w-[SPACE]wijzigen"
1300 print"[SPACE]-v-[SPACE]volgende"
1310 print"[SPACE]-m-[SPACE]MENU"
1320 print"[CRSR-DOWN][SPACE]-?-[SPACE]E]keuze"
1330 geti$:ifi$="" then 1330
1340 if i$=chr$(27) then q=1:return
1350 w=0:forl=1to4:ifi$=mid$("swvm",l,1)
    then l=4:w=1
1360 next l
1370 if w<>1 then 1330
1380 q=0:print"[CRSR-UP][2xCRSR-RIGHT]"
    ;i$:forl=1to300:nextl
1390 if i$="s" then gosub 1460
1400 if i$="w" then gosub 1520
1410 if i$="v" then return
1420 if i$="m" then q=1:return
1430 if q=1 then return
1440 goto1330
1450 rem ----- schrappen -----
1460 e=a:for m=e to g
1470 for r=1 to v
1480 g$(m,r)=g$(m+1,r)
1490 next r:next m
1500 g=g-1:q=1:return
1510 rem ----- wijzigen -----
1520 scnclr
1530 print"[6xCRSR-DOWN]";
1540 for y=1 to v
1550 printv$(y);": :tab(10)g$(a,y)
1560 next y
1570 print"[2xCRSR-DOWN][SPACE]wijzigen[SPACE]of[SPACE]op[SPACE][return][SPACE]drukken":print"[13xCRSR-UP]"
    ;
1580 poke 19,64
1590 for u=1 to v
1600 print"[HOME]"g$(a,u):input"[HOME]"
    ;g$(a,u)
1610 print"[HOME][38xSPACE]"
1620 next u:q=1
1630 poke 19,0:return
1640 rem ----- einde -----
1650 print"[SHIFT CLR][5xCRSR-DOWN][2xSPACE]EINDE:zeker[SPACE]weten?(j/n)"
    "
1660 getf$:iff$="" then 1660
1670 if f$=chr$(27) then return
1680 if f$="n" or f$="N" then return
1690 if f$<>"j" and f$<>"J" then 1660
1700 print"[3xCRSR-DOWN][SPACE]Wilt[SPACE]U[SPACE]niet[SPACE]eerst[SPACE]SAVEN?(j/n)"
    "
1710 gets$:ifs$="" then 1710
1720 if s$=chr$(27) then return

```

```
1730 if s$="j" or s$="J" then return
1740 if s$<>"n" and s$<>"N" then 1710
1750 scncrl:clr:end
1760 rem *****
1770 rem *****
```

** EINDE LISTING mt base c16 **

REGEL	10	249	REGEL	660	250
REGEL	20	160	REGEL	670	172
REGEL	30	81	REGEL	680	142
REGEL	40	122	REGEL	690	176
REGEL	50	116	REGEL	700	232
REGEL	60	45	REGEL	710	228
REGEL	70	245	REGEL	720	120
REGEL	80	214	REGEL	730	97
REGEL	90	62	REGEL	740	84
REGEL	100	136	REGEL	750	186
REGEL	110	78	REGEL	760	136
REGEL	120	150	REGEL	770	84
REGEL	130	24	REGEL	780	230
REGEL	140	174	REGEL	790	33
REGEL	150	152	REGEL	800	143
REGEL	160	197	REGEL	810	226
REGEL	170	231	REGEL	820	142
REGEL	180	94	REGEL	830	151
REGEL	190	106	REGEL	840	10
REGEL	200	0	REGEL	850	214
REGEL	210	219	REGEL	860	144
REGEL	220	249	REGEL	870	161
REGEL	230	65	REGEL	880	179
REGEL	240	89	REGEL	890	62
REGEL	250	116	REGEL	900	229
REGEL	260	132	REGEL	910	212
REGEL	270	214	REGEL	920	123
REGEL	280	6	REGEL	930	226
REGEL	290	86	REGEL	940	142
REGEL	300	17	REGEL	950	82
REGEL	310	26	REGEL	960	232
REGEL	320	76	REGEL	970	250
REGEL	330	75	REGEL	980	53
REGEL	340	4	REGEL	990	167
REGEL	350	159	REGEL	1000	130
REGEL	360	20	REGEL	1010	203
REGEL	370	79	REGEL	1020	89
REGEL	380	146	REGEL	1030	165
REGEL	390	109	REGEL	1040	98
REGEL	400	201	REGEL	1050	85
REGEL	410	159	REGEL	1060	136
REGEL	420	24	REGEL	1070	39
REGEL	430	145	REGEL	1080	112
REGEL	440	195	REGEL	1090	238
REGEL	450	16	REGEL	1100	232
REGEL	460	147	REGEL	1110	219
REGEL	470	254	REGEL	1120	198
REGEL	480	110	REGEL	1130	144
REGEL	490	221	REGEL	1140	25
REGEL	500	24	REGEL	1150	244
REGEL	510	46	REGEL	1160	195
REGEL	520	200	REGEL	1170	244
REGEL	530	142	REGEL	1180	250
REGEL	540	44	REGEL	1190	48
REGEL	550	126	REGEL	1200	154
REGEL	560	181	REGEL	1210	223
REGEL	570	159	REGEL	1220	68
REGEL	580	141	REGEL	1230	172
REGEL	590	195	REGEL	1240	168
REGEL	600	169	REGEL	1250	8
REGEL	610	13	REGEL	1260	28
REGEL	620	89	REGEL	1270	208
REGEL	630	122	REGEL	1280	63
REGEL	640	98	REGEL	1290	245
REGEL	650	136	REGEL	1300	225

REGEL	1310	185	REGEL	1550	96
REGEL	1320	17	REGEL	1560	219
REGEL	1330	164	REGEL	1570	245
REGEL	1340	206	REGEL	1580	151
REGEL	1350	173	REGEL	1590	179
REGEL	1360	206	REGEL	1600	61
REGEL	1370	229	REGEL	1610	240
REGEL	1380	230	REGEL	1620	69
REGEL	1390	64	REGEL	1630	37
REGEL	1400	65	REGEL	1640	120
REGEL	1410	121	REGEL	1650	63
REGEL	1420	222	REGEL	1660	164
REGEL	1430	244	REGEL	1670	93
REGEL	1440	80	REGEL	1680	76
REGEL	1450	93	REGEL	1690	230
REGEL	1460	34	REGEL	1700	138
REGEL	1470	176	REGEL	1710	186
REGEL	1480	155	REGEL	1720	106
REGEL	1490	221	REGEL	1730	94
REGEL	1500	82	REGEL	1740	4
REGEL	1510	77	REGEL	1750	120
REGEL	1520	232	REGEL	1760	165
REGEL	1530	126	REGEL	1770	164
REGEL	1540	183			

Adresbestand C16

Het volgende programma is ook voor de C16. Het is een menugestuurd adressenbestand. Het is verbaasd hoe programmeurs het voor elkaar krijgen om meer met de geheugenruimte van de C16 te doen.

```
1 .adressen laden
2 .adressen save
10 poke53280,0:poke53281,0:poke646,1
15 dimx$(100,4):poke808,225
20 printchr$(147):zx=1
30 printspc(2)"[CTRL 9][CTRL 2]sequen
   tieel[SPACE]adressenbestand[SPACE]
   voor[SPACE]disk"
35 printspc(15)"[CRSR-DOWN][CTRL 9][C
   TRL 2]programmed"
40 printspc(19)"[CRSR-DOWN][CTRL 9][C
   TRL 2]by"
50 printspc(14)"[CRSR-DOWN][CTRL 9][C
   TRL 2]tolsoft[SPACE]inc."
60 printspc(3)"[2xCRSR-DOWN]1.adresse
   n[SPACE]laden"
70 printspc(3)"[CRSR-DOWN]2.adressen[
   SPACE]save"
80 printspc(3)"[CRSR-DOWN]3.uitvoer[S
   PACE]scherm/printer"
90 printspc(3)"[CRSR-DOWN]4.gegevens[
   SPACE]invoeren"
91 printspc(3)"[CRSR-DOWN]5.gegevens[
   SPACE]zoeken"
100 printspc(3)"[CRSR-DOWN]6.gegevens[
   SPACE]wijzigen"
105 printspc(3)"[CRSR-DOWN]7.gegevens[
   SPACE]wissen"
110 printspc(14)"[CRSR-DOWN][CTRL 9][C
   TRL 2]maak[SPACE]uw[SPACE]keuze"
120 getc$:ifc$<"0" or c$>"7" then120
130 on val(c$)gosub140,270,430,890,107
   0,1550,1810
135 goto20
140 rem*****
150 rem* gegevens laden *
160 rem*****
```



```

170 printchr$(147)
175 ifk=2 then i=i+1
176 k=0
180 input "[2xCRSR-DOWN] [SPACE] naam[SPACE] van[SPACE]bestand";m$
185 open15,8,15
190 open1,8,2,m$+"r"
200 input#15,fe
210 iffe<>0then print"[CRSR-DOWN] [3xCRSR-RIGHT]file[SPACE]bestaat[SPACE]niet":fora=1to999:next:goto245
220 ifi=0then i=1
230 input#1,x$(i,1)
231 input#1,x$(i,2)
232 input#1,x$(i,3)
233 input#1,x$(i,4)
240 ifst<>64 theni=i+1:goto230
245 close1:close15:return
270 rem*****
280 rem* gegevens saven *
290 rem*****
295 if i=0then 2070
300 printchr$(147)
310 input "[2xCRSR-DOWN] [SPACE] saven[SPACE] onder[SPACE] filenaam";m$
320 print "[2xCRSR-DOWN] [SPACE] zeker[SPACE] weten[SPACE] (j/n)"
330 getc$:ifc$="j" then350
340 ifc$="n" then return
345 goto330
350 open1,8,2,"0:"m$+"w"
360 fory=1toi
370 print#1,x$(y,1)
371 print#1,x$(y,2)
372 print#1,x$(y,3)
373 print#1,x$(y,4)
380 next
390 close1
420 return
430 rem*****
440 rem* gegevens uitvoer *
450 rem*****
455 if i=0then 2070
460 printchr$(147)
470 printspc(16) "[2xCRSR-DOWN]uitvoer"
480 print "[4xCRSR-DOWN] [9xSPACE]1.naar [SPACE]schem"
490 print "[CRSR-DOWN] [9xSPACE]2.naar[SPACE]printer"
500 getc$:ifc$<"1" or c$>"2"then 500
510 onval(c$)goto520,640
520 x=1
530 printchr$(147)
540 print "[CRSR-DOWN]"spc(16)"adres";x
550 print "[3xCRSR-DOWN] [SPACE]naam";tab(12);x$(x,1)
560 print "[CRSR-DOWN] [SPACE]straat";tab(12);x$(x,2)
570 print "[CRSR-DOWN] [SPACE]postcode";tab(12);x$(x,3)
580 print "[CRSR-DOWN] [SPACE]woonplaats";tab(12);x$(x,4)
590 print "[3xCRSR-DOWN] [2xSPACE]verder gaan[SPACE] (j/n)"
600 getc$:ifc$="" then600
610 ifc$="j" then x=x+1:goto530
620 ifc$="n" then return
630 goto600
640 rem*****
650 rem* gegevens printen *
660 rem*****
665 if i=0then 2070
670 printchr$(147)
680 print "[CRSR-DOWN] [12xSPACE]****pri

```

```

ntent****
690 print "[2xCRSR-DOWN] [CRSR-RIGHT]1.alle[SPACE]adressen[SPACE]in[SPACE]geheugen"
700 print "[CRSR-DOWN] [CRSR-RIGHT]2.een[SPACE]adres[SPACE]naar[SPACE]keuze"
710 getc$:ifc$<"1" or c$>"2" then710
720 on val(c$)goto730,820
730 open4,4
735 gh$=""
740 print#4,chr$(14);tab(13)"adressenlijst"
745 print#4,chr$(15)
750 form=1toi step2
755 j=len(x$(m,1)):s=25-j
760 print#4,x$(m,1);tab(s)x$(m+1,1)
765 j=len(x$(m,2)):s=25-j
770 print#4,x$(m,2);tab(s)x$(m+1,2)
775 j=len(x$(m,3)):s=25-j
780 print#4,x$(m,3);tab(s)x$(m+1,3)
785 j=len(x$(m,4)):s=25-j
790 print#4,x$(m,4);tab(s)x$(m+1,4)
795 print#4,gh$
800 next
810 close4:goto20
820 input "[2xCRSR-DOWN] [2xCRSR-RIGHT]welk[SPACE]adres[SPACE]1,2...";p
830 open4,4
840 print#4,x$(p,1)
850 print#4,x$(p,2)
860 print#4,x$(p,3)
870 print#4,x$(p,4)
880 close4:goto20
890 rem*****
900 rem* gegevens invoeren *
910 rem*****
914 print "[SHIFT CLR] [10xCRSR-DOWN] [SPACE]wilt[SPACE]u[SPACE]adressen[SPACE]invoeren[SPACE]druk[SPACE]dan[SPACE]f-1"
915 getc$:ifc$="" then 915
916 ifc$=chr$(133) then 925
917 goto20
925 i=i+1
930 print "[SHIFT CLR] [7xSPACE]****gegevens[SPACE]invoeren****"
970 print "[CRSR-DOWN] [15xSPACE]adres";i
980 input "[2xCRSR-DOWN] [CRSR-RIGHT]naam";x$(i,1)
990 input "[CRSR-RIGHT]straat";x$(i,2)
1000 input "[CRSR-RIGHT]postcode";x$(i,3)
1010 input "[CRSR-RIGHT]woonplaats";x$(i,4)
1020 print "[3xCRSR-DOWN] [2xCRSR-RIGHT]goed[SPACE]ingevoerd[SPACE] (j/n)"
1030 getc$:ifc$="" then1030
1040 ifc$="j" then1061
1050 ifc$="n" then930
1060 goto1030
1061 print "[7xSPACE]verder[SPACE] (j/n)"
1062 getc$:ifc$="j" then i=i+1:goto930
1063 ifc$="n" then k=2:goto20
1064 goto1062
1070 rem*****
1080 rem* adressen zoeken *
1090 rem*****
1095 if i=0then 2070
1100 printchr$(147)
1105 zx=1
1110 print "[12xSPACE]adressen[SPACE]zoeken"

```

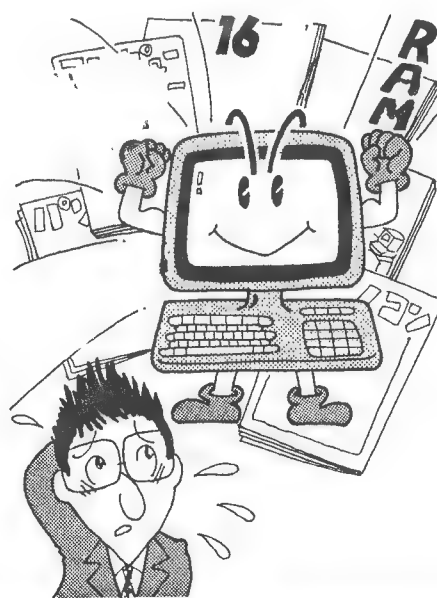
```

1120 print"[3xCRSR-DOWN][17xSPACE]zoeke
n[SPACE]op:"
1130 print"[CRSR-DOWN][16xSPACE][CTRL 9
]n[CTRL 0]aam"
1140 print"[CRSR-DOWN][16xSPACE][CTRL 9
]s[CTRL 0]traat"
1150 print"[CRSR-DOWN][16xSPACE][CTRL 9
]p[CTRL 0]ostcode"
1160 print"[CRSR-DOWN][16xSPACE][CTRL 9
]w[CTRL 0]oonplaats"
1170 input"[3xCRSR-DOWN]maak[SPACE]uw[S
PACE]keuze";z$
1180 ifz$="n" then te=1:goto1310
1190 ifz$="s" then te=2:goto1310
1200 ifz$="p" then te=3:goto1310
1300 ifz$="w" then te=4:goto1310
1305 goto1100
1320 print"[CRSR-DOWN]welke[SPACE]naam,
straat,plaats[SPACE]of[SPACE]postc
ode"
1330 inputzs
1340 print"[SHIFT CLR][11xCRSR-DOWN][7x
CRSR-RIGHT]ik[SPACE]ben[SPACE]even
[SPACE]aan[SPACE]het[SPACE]zoeken"
1350 forc=zxtol
1360 poke56,207:ifz$=x$(c,te) then 1380
1370 next
1375 iffl=5 then1540:print"[CRSR-DOWN][
10xSPACE]adres[SPACE]niet[SPACE]ge
vonden":goto1450
1380 printchr$(147)
1390 print"het[SPACE]adres[SPACE]dat[SP
ACE]door[SPACE]u[SPACE]gezocht[SPA
CE]werd[SPACE]is:"
1400 print"[2xCRSR-DOWN][15xSPACE]adres
";c
1410 print"[3xCRSR-DOWN][CRSR-RIGHT]naa
m";tab(12)x$(c,1)
1420 print"[CRSR-DOWN][CRSR-RIGHT]straa
t";tab(12)x$(c,2)
1430 print"[CRSR-DOWN][CRSR-RIGHT]postc
ode";tab(12)x$(c,3)
1440 print"[CRSR-DOWN][CRSR-RIGHT]woonp
laats";tab(12)x$(c,4)
1445 ifc<>i then gosub1500
1450 print"[2xCRSR-DOWN][10xSPACE]verde
r[SPACE]zoeken[SPACE](j/n)"
1460 getc$:ifc$="j" then 1070
1470 ifc$="n" then 20
1480 goto1460
1500 print"[2xCRSR-DOWN][10xSPACE]verde
r[SPACE]zoeken[SPACE](j/n)":fl=5
1510 getc$:ifc$="j" then zx=c+1:goto135
0
1520 ifc$="n" then 20
1530 goto1510
1540 print"[SHIFT CLR][8xCRSR-DOWN][3xS
PACE]geen[SPACE]adressen[SPACE]mee
r[SPACE]met[SPACE]dit[SPACE]gegeve
n":goto1450
1550 rem*****
1560 rem* gegevens wijzigen *
1570 rem*****
1571 if i=0then 2070
1575 print"[SHIFT CLR][3xSPACE]wilt[SPA
CE]u[SPACE]werkelijk[SPACE]wijzige
n[SPACE](j/n)
1576 get c$:if c$="j" then 1580
1577 if c$="n" then 20
1578 goto 1576
1580 printchr$(147)
1590 print"[11xSPACE][CTRL 9][CTRL 2]ad
ressen[SPACE]wijzigen[CTRL 0]"
1600 input"[3xCRSR-DOWN][SPACE]welk[SPA
CE]nummer";na
1610 print"naam[SPACE]is[SPACE]";x$(na,
1)
1620 print"adres[SPACE]is[SPACE]";x$(na
,2)
1630 print"postcode[SPACE]is[SPACE]";x$
(na,3)
1640 print"woonplaats[SPACE]is[SPACE]";
x$(na,4)
1650 print:input"nieuwe[SPACE]naam[SPAC
E]";x$(na,1)
1660 input"nieuw[SPACE]adres[SPACE]";x$
(na,2)
1670 input"nieuwe[SPACE]postcode";x$(na
,3)
1680 input"nieuwe[SPACE]woonplaats";x$(
na,4)
1690 print"[2xCRSR-DOWN][9xSPACE]gegeve
ns[SPACE]goed[SPACE](j/n)"
1700 get c$:if c$="j" then 1730
1710 if c$="n" then 1650
1720 goto 1700
1730 print"[CRSR-DOWN]nog[SPACE]meer[SP
ACE]adressen
1740 get c$:if c$="j" then print"[SHIFT
CLR]";goto 1580
1750 if c$="n" then 1770
1760 goto 1740
1770 open1,8,15,"s:"+m$
1780 close1
1790 gosub 350
1800 goto 20
1810 rem*****
1820 rem* gegevens wissen *
1830 rem*****
1835 if i=0then 2070
1840 printchr$(147);"[CRSR-DOWN][SPACE]
u[SPACE]wilt[SPACE]echt[SPACE]wiss
en[SPACE](j/n)"
1850 get c$:if c$="j" then 1880
1860 if c$="n" then return
1870 goto 1850
1880 printchr$(147)
1890 print"[3xCRSR-DOWN][SPACE]geef[SPA
CE]nummer[SPACE]te[SPACE]wissen[SP
ACE]adres"
1900 input nr
1910 print"[3xCRSR-DOWN][SPACE]naam[SPA
CE]";x$(nr,1)
1920 print"[SPACE]adres[SPACE]";x$(nr,2
)
1930 print"[SPACE]postcode[SPACE]";x$(n
r,3)
1940 print"[SPACE]woonplaats[SPACE]";x$
(nr,4)
1950 print"[2xCRSR-DOWN][2xSPACE]goede[
SPACE]adres"
1960 get c$:if c$="n" then 1840
1970 ifc$="j" then 1990
1980 goto 19601990for a=nr+1 to i
2000 x$(a-1,1)=x$(a,1)
2010 x$(a-1,2)=x$(a,2)
2020 x$(a-1,3)=x$(a,3)
2030 x$(a-1,4)=x$(a,4)
2040 next
2050 for a=1 to 4:x$(i,a)-"":next:i=i-1
2060 goto 1770
2070 printchr$(147)
2080 print"[11xCRSR-DOWN]";spc(3)"er[SP
ACE]is[SPACE]geen[SPACE]bestand[SP
ACE]in[SPACE]de[SPACE]computer"
2090 for c=1 to 500:next:goto 20

```

** EINDE LISTING adressenbestand C16**

REGEL	1	231	REGEL	580	32	REGEL	1160	154	REGEL	1660	181
REGEL	2	101	REGEL	590	7	REGEL	1170	255	REGEL	1670	237
REGEL	10	243	REGEL	600	103	REGEL	1180	248	REGEL	1680	149
REGEL	15	122	REGEL	610	113	REGEL	1190	254	REGEL	1690	138
REGEL	20	28	REGEL	620	107	REGEL	1200	252	REGEL	1700	230
REGEL	30	0	REGEL	630	31	REGEL	1300	4	REGEL	1710	169
REGEL	35	40	REGEL	640	215	REGEL	1305	75	REGEL	1720	81
REGEL	40	217	REGEL	650	87	REGEL	1320	101	REGEL	1730	46
REGEL	50	108	REGEL	660	215	REGEL	1330	3	REGEL	1740	28
REGEL	60	25	REGEL	665	38	REGEL	1340	10	REGEL	1750	172
REGEL	70	34	REGEL	670	77	REGEL	1350	21	REGEL	1760	85
REGEL	80	148	REGEL	680	94	REGEL	1360	4	REGEL	1770	222
REGEL	90	12	REGEL	690	205	REGEL	1370	130	REGEL	1780	209
REGEL	91	115	REGEL	700	88	REGEL	1375	31	REGEL	1790	37
REGEL	100	15	REGEL	710	217	REGEL	1380	77	REGEL	1800	235
REGEL	105	130	REGEL	720	247	REGEL	1390	171	REGEL	1810	173
REGEL	110	131	REGEL	730	51	REGEL	1400	236	REGEL	1820	16
REGEL	120	216	REGEL	735	169	REGEL	1410	33	REGEL	1830	173
REGEL	130	102	REGEL	740	255	REGEL	1420	178	REGEL	1835	38
REGEL	135	235	REGEL	745	118	REGEL	1430	69	REGEL	1840	135
REGEL	140	131	REGEL	750	121	REGEL	1440	237	REGEL	1850	236
REGEL	150	155	REGEL	755	34	REGEL	1445	117	REGEL	1860	107
REGEL	160	131	REGEL	760	27	REGEL	1450	171	REGEL	1870	87
REGEL	170	77	REGEL	765	35	REGEL	1460	227	REGEL	1880	77
REGEL	175	128	REGEL	770	29	REGEL	1470	63	REGEL	1890	220
REGEL	176	45	REGEL	775	36	REGEL	1480	84	REGEL	1900	37
REGEL	180	154	REGEL	780	31	REGEL	1500	94	REGEL	1910	50
REGEL	185	251	REGEL	785	37	REGEL	1510	41	REGEL	1920	82
REGEL	190	155	REGEL	790	33	REGEL	1520	63	REGEL	1930	69
REGEL	200	161	REGEL	795	171	REGEL	1530	80	REGEL	1940	237
REGEL	210	50	REGEL	800	130	REGEL	1540	234	REGEL	1950	210
REGEL	220	137	REGEL	810	249	REGEL	1550	1	REGEL	1960	236
REGEL	230	84	REGEL	820	107	REGEL	1560	158	REGEL	1970	172
REGEL	231	85	REGEL	830	51	REGEL	1570	1	REGEL	1980	89
REGEL	232	86	REGEL	840	114	REGEL	1571	38	REGEL	1990	220
REGEL	233	87	REGEL	850	115	REGEL	1575	10	REGEL	2000	100
REGEL	240	30	REGEL	860	116	REGEL	1576	233	REGEL	2010	102
REGEL	245	217	REGEL	870	117	REGEL	1577	63	REGEL	2020	104
REGEL	270	131	REGEL	880	249	REGEL	1578	92	REGEL	2030	106
REGEL	280	180	REGEL	890	1	REGEL	1580	77	REGEL	2040	130
REGEL	290	131	REGEL	900	157	REGEL	1590	66	REGEL	2050	70
REGEL	295	38	REGEL	910	1	REGEL	1600	205	REGEL	2060	88
REGEL	300	77	REGEL	914	23	REGEL	1610	138	REGEL	2070	77
REGEL	310	201	REGEL	915	112	REGEL	1620	221	REGEL	2080	251
REGEL	320	27	REGEL	916	154	REGEL	1630	208	REGEL	2090	193
REGEL	330	179	REGEL	917	235	REGEL	1640	120			
REGEL	340	107	REGEL	925	31	REGEL	1650	122			
REGEL	345	31	REGEL	930	122						
REGEL	350	248	REGEL	970	225						
REGEL	360	170	REGEL	980	211						
REGEL	370	120	REGEL	990	100						
REGEL	371	121	REGEL	1000	247						
REGEL	372	122	REGEL	1010	159						
REGEL	373	123	REGEL	1020	2						
REGEL	380	130	REGEL	1030	149						
REGEL	390	209	REGEL	1040	161						
REGEL	420	142	REGEL	1050	121						
REGEL	430	215	REGEL	1060	77						
REGEL	440	101	REGEL	1061	189						
REGEL	450	215	REGEL	1062	153						
REGEL	455	38	REGEL	1063	49						
REGEL	460	77	REGEL	1064	82						
REGEL	470	99	REGEL	1070	173						
REGEL	480	100	REGEL	1080	4						
REGEL	490	148	REGEL	1090	173						
REGEL	500	214	REGEL	1095	38						
REGEL	510	244	REGEL	1100	77						
REGEL	520	59	REGEL	1105	149						
REGEL	530	77	REGEL	1110	254						
REGEL	540	106	REGEL	1120	181						
REGEL	550	84	REGEL	1130	175						
REGEL	560	229	REGEL	1140	97						
REGEL	570	120	REGEL	1150	243						



Desktopvideo is het zonder een uitgebreide studio vervaardigen van videoproducties. Gewoon op het bureau dus. Vroeger kwam dat neer op wat ruw montagewerk met twee videorecorders. Tegenwoordig gaat alles echter snel, comfortabel en (semi)-professioneel met behulp van videocomputers zoals de Amiga, video-software, titel- en Special Effect Generatoren, digitizers, genlockers en videoprocessors/correctors.

Desktop Video

Complete videoproducties met de Amiga, video-processor/corrector en SEG

Desktop en video zijn voor menige zakelijke gebruiker van video-presentaties en video-professional een bedenkelijke combinatie. Voor het maken van videoproducties die klinken als een klok is immers een goed uitgeruste studio nodig? En wat "klungelen" achter bureau kan toch nooit meer opleveren dan slordig gemonteerde familiefilms? Beide vooroordelen zijn inmiddels door de techniek achterhaald. Voor relatief weinig geld staat nu een volledig uitgeruste video-studio in de huis- of hobbykamer. Iedere video-enthousiast kan daarmee in korte tijd (semi)-professionele montages en trukages mee maken.

Video-producties zijn momenteel erg gewild bij bedrijfspresentaties. Geanimeerde beelden zeggen immers veel meer dan saai jaarcijfers, marktontwikkelingen en produkt-folders. Toch zagen vele kleinere bedrijven van videoproducties af, omdat uitbesteden en in house-producties gewoon te duur waren. Met name in de beginnende jaren van de video waren professionele montage en trucage voor de kleine videogebruiker gewoon onbetaalbaar. Vakstudio's die vele tientallen, en dikwijls enige honderd-, duizenden gulden geïnvesteerd hadden maakten de dienst uit. Degenen met een krap budget moesten zelf maar wat aanfrutelen met een slaaf- en moederrecorder.

De laatste jaren is deze situatie gelukkig flink veranderd. Video werd snel populair en de vraag naar presentaties voor zakelijk, onderwijs, reclame en hobby-gebruik nam dienovereenkomstig een komstigeenkomstigeenkomstigeenkomstigeenkomstig toe. Nieuwe elektronische technieken deden hun intrede en de apparatuurprijzen kwamen binnen het bereik van de

kleine gebruiker. Er zijn nu twee mogelijkheden:

1. Het gebruik van een grote video-studio voor het maken van topproducties. Hiermee zijn uiteraard grote bedragen van vaak meer dan 100.000 gulden gemoeid.
2. Kleinschalige video-producties, **desktopvideo**, met een uitrusting naar budget. De kwaliteit is misschien iets minder, maar wel heel bruikbaar en zeker acceptabel voor de beoogde doelgroepen.

Desktopvideo-configuraties

Welke uitrusting het beste bij u past hangt uiteraard van de gestelde eisen en het beschikbare budget af. Arbitrair zijn er vier verschillende niveaus te onderscheiden:

- De **eenvoudige beginnersconfiguratie** bestaande uit een videocassetterecorder (VCR) en een daaraan gekoppelde huis- of personal computer. De computer wordt hierbij gebruikt als titelgenerator, animator, en muziekstudio.

- De **intermediate-configuratie** breidt de beginnersset uit met een videocamera. De met de videocamera gemaakt beelden kunnen samen met de door de computer gemaakte graphics (tekeningen, animaties en titels) op een VCR gemonteerd worden. Dit intermediate desktopvideo systeem is heel geschikt voor het maken van vakantie-, familie- of educatieve video's.
- Het **kleine zaken-systeem** omvat naast de VCR, videocamera en computer een genlocker voor het maken van superposities (twee videobeelden over elkaar heen leggen), een digitizer voor het overnemen van grafische afbeeldingen of foto's en een kopieerstandaard. Een aantal producenten zal in dit systeem ook een Special Effect Generator (SEG) en een tweede VCR willen opnemen. Met behulp van deze videoconfiguratie zijn bescheiden, maar professioneel ogende videoproducties in betrekkelijk korte tijd te maken. Het kleine zaken systeem is geschikt voor

de zakelijke gebruiker, onderwijsman/vrouw, serieuze hobbyist en ook voor kleine reclame-producties.

- De **complete desktop-studio** omvat naast de bij het kleine zakensysteem omschreven uitrusting een video-editor, meer dan twee VCR's (een moeder- en enkele slaafrecorders) en een SEG. Kortom de complete uitrusting voor commercials, demonstratie-, onderwijs- en zakelijke presentatie-video's. Ook de veel eisende hobbyist waant zich met een dergelijke configuratie in het video- Mekka. Helaas staat er wel meer dan f 10.000 gulden onder aan het budget.

Ondanks alle mogelijkheden die de genoemde configuraties bieden kunnen zij nooit de eigen visie en de creativiteit van de videomaker vervangen!

De hardware slecht alleen de grenzen der mogelijkheden. De maker blijft bepalend voor het eindprodukt en professionele resultaten zijn ook met een heel eenvoudige opzet mogelijk.

De videocomputer

Een videocomputer behoeft in wezen geen peperduur en specialistisch apparaat te zijn. Met de juiste software aan boord valt van een Amiga 500, 1000 of 2000 computer met een kleuren-monitor een handige en creatieve videohulp te maken. De flitsende animaties, meer dan 4.000 schermkleuren, speciale TV- titelmogelijkheden, muziekstudio-mogelijkheden, hedenhedenhedenheden HIRES-beelden en 3D-tekenpakketten zijn inmiddels we-reldberoemd geworden. Als bezwaren voor de Amiga gelden de relatief dure randapparatuur en de echte professional vindt de videokwaliteit nog net niet goed genoeg. NB.: Commodore is inmiddels drukdoende de **PAL-videokwaliteit** van de Amiga 2000 op te vijzelen. In het najaar worden een speciale genlocker- en Composiet video-kaart verwacht. Tot dan zit de videohobbyist nog aardig in het kwaliteitsschip. De in de 1000 gebouwde PAL-encoder en de TV/composiet-modulator voor de 500 bieden onvoldoende PAL-kwaliteit voor (semi)-professionele doeleinden. De speciale PAL-kaart voor de 2000 brengt het er in deze iets beter van af. In RGB of op een PAL-composiet-video lockende monitor ziet het er alle-

maal gedetailleerd en kleurrijk uit. Bij opname op standaard VHS-formaat wordt het de door de huiskamer TV verwendende gebruiker echter streperig en groen en geel voor de ogen. Een kopie is al helemaal niet meer om aan te zien. Kortom het composiet-videosignaal van computer-PAL-encoders is niet voldoende voor professionele VHS-kwaliteit. Wie normale broadcasting quality wil zal diep in de beurs voor een locked PAL-encoder of professionele genlocker moeten tasten.

Voor de amiga is inmiddels voldoende videosoftware verkrijgbaar. Wij noemen hier: de Deluxe Video 1.2, Deluxe Paint II en Aegis Animator.

Processors en correctors

Bij de opname, het overspelen of kopiëren, het maken van trucages en het monteren ontstaan onvermijdelijk beeldfouten, kleurverschuivingen en/of ruis. Zoals bekend loopt bij het **kopiëren** de scherpte terug en ontstaat ruis. Bij de opname worden nogal eens de **witbalans** of de **tegenlicht-**

hetzelfde met de video-signalen: elimineert de ruis en bevoordeelt de beeldkwa-liteit bepalende frequenties.

Het verschil tussen een corrector en een processor is tegenwoordig niet zo duidelijk meer. Vroeger richtte de processor of enhancer zich voornamelijk op de overspeel- en de corrector zich op de opnamefouten. Bij de nieuwste modellen ziet men steeds meer functies van de een in de ander geïntegreerd worden.

Videoprocessoren en correctors bieden in het algemeen de volgende mogelijkheden:

- **Verbetering van de kopieer-kwaliteit.** Verlies aan beeldscherpte, contrast en de toename van ruis worden zoveel mogelijk tegengegaan. Enkele apparaten bieden de mogelijkheid om meerdere kopieën tegelijk te maken.
- **Correcties van het inputsignaal.** Bijvoorbeeld het traploos bijstellen van de kleurbalans met een joystick, het oppeppen van contrast-



voorbeelden van wipes

correctie vergeten. En bij het monteren met slaaf- en moeder-VCR ontstaan nogal eens storende strepen. Een **goede videoband** in de recorder stoppen helpt wel enigszins bij het kopiëren, maar na drie maal overschrijven wordt het resultaat bij consumenten-video al gauw bedroevend. De elektronische vervuiling wint het van de tape- en recordertechniek. Hulp komt er in de vorm van de microprocessor gestuurde **video-processor** en **videocorrector**. Net als bij uw stereo-installaties is het via elektronische foefjes, digitale correctie en filteringen mogelijk om stoor- en ruissignalen te elimineren.

Een processor of corrector doet

arme beelden, het verhogen/verlagen van de kleurverzadiging, verbeteren van de scherpte (lukt niet bij echt onscherpe opnamen!) en het corrigeren van onder/overbelichting (voor zover het signaal nog te bewerken valt).

- **Verbetering van de geluidskwaliteit en geluidsmixages.** De meeste apparaten beperken zich tot de mixages. Een enkeling beschikt ook over audiofilteringen.
- Enkele apparaten beschikken ook over beperkte trukage- mogelijkheden zoals fade ins en -outs, wipes, valse kleuren en sound on sound.

- **Screen splitting.** De monitor wordt als het ware in tweeën gedeeld waarbij de linker helft het beeld voor en de rechter helft het beeld na bewerking laat zien.
- **Bypass-optie** om signalen die geen bewerking behoeven ongehinderd te laten passeren. U kunt de processor of corrector dan standaard in uw videosysteem opnemen.

Enkele voorbeelden van videoprocessors -correctors zijn:

* De **Elac VHP-185** video-processor regelt de helderheid, de kleurbalans (niet de fase!) en de scherpte bij. Verder beschikt deze processor over een fadermogelijkheid en een regelaar voor de geluidssterkte. De prijs ligt rond de f 600,-.

* De **Bauer VED-200** biedt dezelfde mogelijkheden als de Elac met als extra's gemakkelijk werkende schuifregelaars i.p.v. draaiknoppen en screen splitting voor de monitor. De VED-200 kost circa f 1.100,-.

* De **JVC JX-E3** Video-processor werd speciaal ontworpen om de kopieer-kwaliteit te verbeteren. Detailverbetering, verbetering van de totaal-scherpte via de techniek van de dubbele vertragsingslijn en een continu instelbaar ruisfilter zijn de grote troeven van deze processor.

* De **JVC JX-C7 videocorrector** beschikt over een praktische kleurstelling (rood, groen en blauw) via een traploos verplaatsbare joystick, regelbare kleurverzadiging, beeldverbetering bij het kopiëren via de techniek van de dubbele vertragsingslijn, ruisfilter, stabilisatie-circuit voor schone video-overgangen, audio/videofaders en de mogelijkheid tot geluidmixen. Voor prijzen van de JVC-produkten uw dealer.

* De **Kramer VS 11 E MK 2 video-processor** biedt o.a. video gain-regeling, contrastregeling, kleurregeling, faseregeling, scherptecorrectie, screen splitter, een 3-voudige ingangsschakeling, ruisfiltering, 3 gebufferde AV-stereoingangen, on-line videoprocessing en een 10-polige camera-aansluiting.

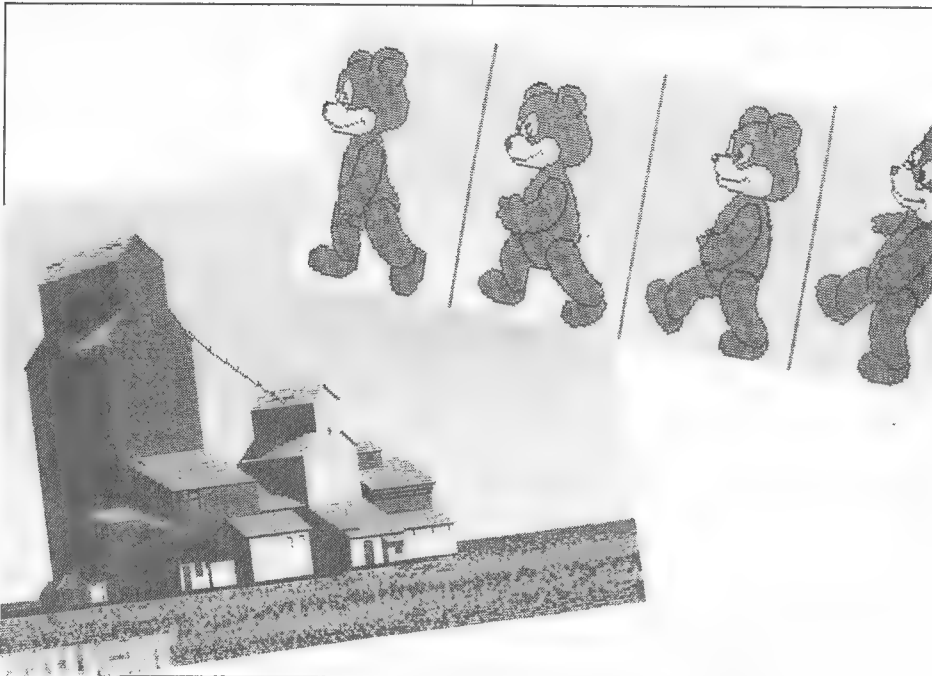
SEG's

Special Effect Generators zijn, zoals de naam al aangeeft, bedoeld, voor het maken van videotrukages. De trukages worden dus **achteraf** aangebracht. Dat werkt meestal handiger dan tijdens de opname (fade ins/outs en het gebruik van kleurenfilters). Vooral als er nog gemonteerd moet worden.

SEG's worden net als filtersystemen nogal eens sceptisch bekeken. Een veel gehoord vooroordeel verkondigt dat al die trukages slechts gebruikt worden om het gebrek aan talent en creativiteit van de maker te verhullen.

Men vergeet daarbij echter twee belangrijke zaken:

1. Een SEG wordt niet uitsluitend effectmatig gebruikt, maar ook ingezet voor het maken van nette beeldovergangen zoals wipes, fade in en fade out.



2. Mits doeltreffend gebruikt kunnen de Special Effects de video ook professioneel aankleden of de boodschap van de film accentueren.

Overdaad schaadt, maar bij het geheel weglaten van trukages mist de videomaker heel wat creatieve mogelijkheden en montageplezier!

Het zou te ver voeren om hier alle mogelijkheden van de SEG's te bespreken. Wij beperken ons daarom tot de meest gebruikte trukages:

- **Fade in/out.** Overgangen van donker naar licht (fade in) en van licht naar donker (fade out) zijn een geliefde montage-methode. Met een SEG zijn zij dikwijls doeltreffender aan te brengen dan bij de opname. Een opname kan bijvoorbeeld te lang blijken en met de SEG kunt u een flink stuk voor het einde beginnen met faden.
- **Wipe-effecten.** Bij een wipe wordt het life videobeeld geleidelijk aan vervangen door een effen (gekleurd) vlak. Daarbij zijn vele varianten zoals vierkanten, circels, rechthoeken en diagonalen moge-

lijk. Soms kunnen meerdere wipetypen met elkaar of fades gecombineerd worden en bestaat de mogelijkheid tot een reverse wipe. Het wipen kan doorgaans zowel automatisch (voor ingestelde tijd) als met de hand (meer mogelijkheden, maar niet zo vloeiend).

- **Colorize.** Dit effect lijkt wel wat op het gebruik van een gekleurd filter. Het beeld krijgt een eigen sterk overheersende kleur. Bijvoorbeeld blauw als suggestie van koude. Er

is meestal keuze uit meerdere kleuren en een traploos instelbare intensiteit.

- **Paint** geeft het effect van gedigitaliseerde computerbeelden of een pseudo-solarisatie.
- **Negatief-schakeling** voor het omkeren van het positieve videobeeld tot science fiction achtige creaties.
- **Beeld in beeld- en superimpose-technieken.** Hierbij wordt het beeld van een videobron over dat van een ander heen gelegd. Het eenvoudigste geval is een titel die later over de eigenlijke video-opname wordt aangebracht. Bij geavanceerde SEG's zijn zelfs met de joystick verplaatsbare beeldkaders en **overvloei-ers** mogelijk.
- **Auto cross input.** Een wipevariant tussen twee videorecorders. Zodra het beeld van recorder 1 door de wipe verdwenen is komt dat van VCR2 op.
- **Audio mixages en sound on sound** met input van de huiskamerstereo.

- **Zwart-wit-optie.**
- **Defocus** of softeffect voor een dromerige onscherpte.

SEG's krijgen steeds meer mogelijkheden. Veelzijdige modellen zijn jammer genoeg nog altijd flink aan de prijs. Als voorbeeld worden hier een wat beperkter en een uitgebreider model kort besproken:

* De **JVC JX-W9** is een SEG van rond de f 1500,- met verrassend veel mogelijkheden. O.a. wipes in 16 patronen en 6 kleuren, auto cross input, fade in/out, zwartwit, paint, negatief, colorize, wipe reverse, diverse effectcombinaties en audiomix.

* De **Technofilm SEG** kost rond de f 3000,-, maar biedt dan ook wel wat meer dan de JVC SEG. Vier video-bronnen i.p.v. twee, een pattern generator voor 60 verschillende wipes, fade in/out, polarisation (zachte kleurtinten) en posterisation (negatieve beelden) met instelbare intensiteit, joystick bestuurbare beeld in beeld-techniek en spotlight techniek om het basisbeeld naar de achtergrond te drukken (= minder helder weergeven), aanpassing van de grootte van het beeld-kader en audio-breakaway voor het onge-stoord doorlopen van het geluid bij beeldtrucages.

Overige randapparatuur

Het naadloos, dat wil zeggen zonder storende beeldlijnen en ruis, monteren van video-opnamen is geen sinecure. Een **editor** kan daarbij goede diensten bewijzen. Editors zijn geavanceerde montage-apparaten. Zij plaatsen de VCRs on line, starten de recorders, controleren de bandsnelheid, synchro-niseren de beelden en kunnen de start en eindposities van de verschillende scènes tot op 1 beeld nauwkeurig in hun geheugen opslaan. De gebruiker behoeft slechts de juiste tellerstanden plus volgorde in te voeren en op de starknop te drukken. De editor doet de rest.

Editors zijn o.a. verkrijgbaar van Kramer, Sony en Canon.

Een computer kan niet zo maar een foto of illustratie bewerken. De afbeelding moet eerst gescand (in beeldlijnen ontleed) en vervolgens in bits en bytes omgezet worden. Dat doet een zogenaamde **digitizer** voor u. Is het beeld eenmaal gedigitaliseerd dan kan de videoproducent(e) dit met een geschikt teken- of paintprogramma inkleuren, dingen erbij tekenen, vervreemden, roteren, dupliceren vergroten en verkleinen. Kortom de gehele

grafische trukendoos op het gedigitaliseerde beeld loslasten.

Bij video gaat het meestal om met camera's opgenomen beelden. Het beeld wordt door de hardware, bijvoorbeeld DigiView voor de Amiga, gedigitaliseerd en vervolgens met paintsoftware (in ons voorbeeld Digi-Paint) bewerkt.

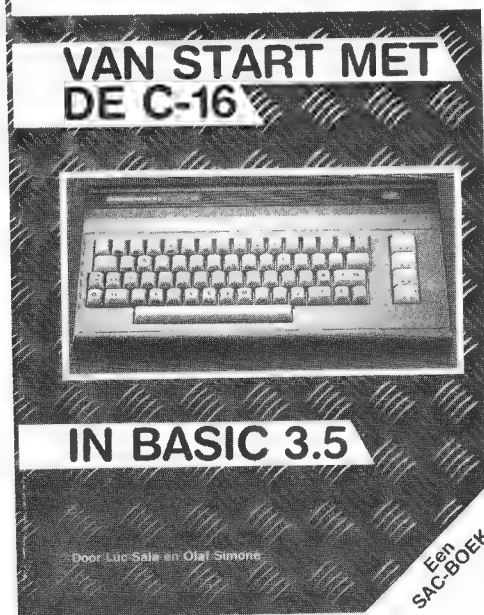
Voor het goed overnemen van te digitaliseren beelden is een copystand met bijbehorende egale verlichting onmisbaar. Deze copystand is tevens bruikbaar voor het filmen van tabletps. Professionele videoproducties liggen tegenwoordig binnen ieders bereik. Voor relatief weinig geldt koopt men bij de eigen PC of 68.000- machine al een complete studio voor het maken van familiefilms, reclameproducties, zakelijke presentaties en educatieve of promotievideo's. De laatste grenzen worden nog technisch geslecht en dan houdt niets de eigen creativiteit meer tegen. Alleen zij die topkwaliteit nastreven zullen nog naar de grote professionele studio's moeten uitwijken.

Een Nederlandstalig boek (168 pag.), waarmee u niet alleen inzicht krijgt in wat er met Basic 3.5 allemaal mogelijk is, maar ook veel bredere computertoepassingen behandeld worden en randapparatuur en software uitgebreid aan de orde komen.

Het onmisbare boek voor iedere C-16 bezitter, compleet met vele programmavoorbeelden, zoals een klein tekstbewerkingsprogramma.

Direkt te bestellen bij: INFOLIST

Van start met de C-16 in Basic 3.5



**VOOR DE C-16
ook bruikbaar
voor PLUS/4**

Prijs

f 27,50

**plus f 3,-
verzendskosten**

Postbus 1047

1270 BA Huizen

tel.: 02152-62343

Alleen levering na vooruitbetaling
van f 30,50.

Gironummer 3157656 tnv. Infolist, Huizen.

In twee eerdere afleveringen heeft Peter Heinckiens verteld over de mogelijkheden tot het maken van een 80-kolomsscherm op de C 64. Hij heeft daarvoor ook een programma geschreven, Minigraph, dat helaas vorige maal niet volledig kon worden geplaatst. Hieronder volgt daarvan het staartje in de vorm van listing 3 en 4.

Minigraph (slot)

Listing 3 & 4.

```

10 REM =====
11 REM
12 REM             MINIGRAPH      -      DEMO 2
13 REM
14 REM             DOOR
15 REM
16 REM             PETER HEINCKIENS
17 REM
18 REM             (C) 1986
19 REM
20 REM =====
21 :
22 FAST
23 :
24 REM  PARAMETERS
25 AX = 255 : BX = 256
26 KX = 62 : YK = 100
27 PX = DEC("BBB")
28 UX = DEC("BBB")
29 YAK = DEC("009B")
30 XAK = DEC("000D")
31 :
32 REM  BITMAP AAN
33 SYS DEC("0018")
34 :
35 GOTO 48
36 :
37 REM  TEKEN HORIZONTALE AS
38 POKE DEC("1153"), X2 AND AX
39 POKE DEC("1154"), X2 / BX
40 SYS XAK, X1 AND AX, X1 / BX, Y
41 RETURN
42 :
43 REM  TEKEN VERTIKALE AS
44 POKE DEC("1153"), Y2
45 SYS YAK, X AND AX, X / BX, Y1
46 RETURN
47 :
48 REM  HOOFDPROGRAMMA
49 :
50 Y=30 : X1=100 : X2=400 : GOSUB 37
51 Y=150 : X1=100 : X2=400 : GOSUB 37
52 X=100 : Y1=30 : Y2=150 : GOSUB 43
53 X=400 : Y1=30 : Y2=150 : GOSUB 43
54 :
55 FOR Y=31 TO 149
56 X1=101 : X2=399
57 GOSUB 37
58 NEXT
59 :
60 FOR X=50 TO 450
61 Y = 150 - (X/20 - 12.5)*2
62 IF X>99 AND X<401 THEN SYS UX, X AND AX, X / BX, Y
63 IF X<99 OR X>401 THEN SYS PX, X AND AX, X / BX, Y
64 NEXT
65 :
66 REM  KLEUR
67 FOR A=1 TO 15
68 FOR V=1 TO 15

```

LISTING 3

```

MONITOR
PC SR AC XR YR SP
; FB000 00 00 00 00 00 F8

```

```

>000000 0C 00 06 2C 00 06 10 FB 8D 01 06 60 8C 00 06 2C:XXXXXXXXXX
>000010 00 06 10 FB AD 01 06 60 A0 19 A9 87 20 00 08 20:XXXXXXXXXX
>000020 22 08 A2 40 A0 12 8A 20 00 08 A0 1F A9 00 20 00:XXXXXXXXXX
>000030 08 A0 1E 20 00 08 CH 10 EB 60 A0 19 A9 40 20 00:XXXXXXXXXX
>000040 08 20 0C CE A2 38 4C DC E1 85 FA 86 FB 84 FC 48:XXXXXXXXXX
>000050 20 7D 08 68 38 29 07 85 FA A9 07 E5 FA A8 A9 01:XXXXXXXXXX
>000060 00 00 F0 04 0A 88 00 FC 85 FA A0 12 A5 FD 20 00:XXXXXXXXXX
>000070 08 08 A5 FC 20 00 08 A0 1F 20 0C 08 60 46 FB 66:XXXXXXXXXX
>000080 FA 46 FB 66 FA 46 FB 66 FA 18 A9 00 85 FD A5 FC:XXXXXXXXXX
>000090 A0 04 65 FC 90 03 E6 FD 18 88 D0 F6 0A 26 FD 0A:XXXXXXXXXX
>0000A0 26 FD 0A 26 FD 0A 26 FD 65 FA 85 FC A5 FB 65 FD:XXXXXXXXXX
>0000B0 85 FD 60 20 49 08 05 FA 4C C6 08 20 49 08 85 FB:XXXXXXXXXX
>0000C0 A5 FA 49 FF 25 FB 48 A0 12 A5 FD 20 00 08 C8 A5:XXXXXXXXXX
>0000D0 FC 20 00 08 68 A0 1F 20 00 08 60 A0 50 11 8D 52:XXXXXXXXXX
>0000E0 11 A0 51 11 8D 53 11 A0 00 84 FE A0 12 AD 53 11:XXXXXXXXXX
>0000F0 20 00 08 C8 AD 52 11 20 00 08 A0 1F 20 0C 08 A4:XXXXXXXXXX
>000100 FE 91 FA 18 AD 52 11 69 50 8D 52 11 90 03 EE 53:XXXXXXXXXX
>000110 11 C8 C0 C8 D0 03 A6 FC D0 04 A2 07 10 63 CA D0:XXXXXXXXXX
>000120 12 A0 00 B1 FA 4A 91 FA C8 C0 C8 D0 F6 CE 50 11:XXXXXXXXXX
>000130 4C 81 0C CE 50 11 AD 50 11 8D 52 11 AD 51 11 8D:XXXXXXXXXX
>000140 53 11 A0 00 84 FE A0 12 AD 53 11 20 00 08 C8 AD:XXXXXXXXXX
>000150 52 11 20 00 08 A0 1F 20 0C 08 85 FD A4 FE 46 FD:XXXXXXXXXX
>000160 B1 FA 6A 91 FA CA D0 F6 4A 91 FA 18 AD 52 11 69:XXXXXXXXXX
>000170 50 8D 52 11 90 03 EE 53 11 A6 FC CA C8 C0 C8 D0:XXXXXXXXXX
>000180 C3 86 FC 60 EA A0 00 B1 FA 09 80 20 D2 FF 20 D2:XXXXXXXXXX
>000190 FF C8 C0 C8 D0 F1 60 EA A9 5C 8D 54 11 A9 4F 8D:XXXXXXXXXX
>0001A0 50 11 A9 00 8D 51 11 A9 00 85 FC A9 00 85 FA A9:XXXXXXXXXX
>0001B0 04 85 FB 20 D8 08 A9 00 85 FA A9 04 85 FB 20 85:XXXXXXXXXX
>0001C0 0C A9 0D 20 D2 FF CE 54 11 D0 E0 60 EA 8D 50 11:XXXXXXXXXX
>0001D0 8E 51 11 8D 52 11 29 07 F0 36 85 28 A9 07 38 E5:XXXXXXXXXX
>0001E0 28 85 28 AD 50 11 AE 51 11 AC 52 11 20 83 08 AD:XXXXXXXXXX
>0001F0 50 11 69 01 8D 50 11 90 03 EE 51 11 AD 51 11 CD:XXXXXXXXXX
>000200 54 11 D0 08 AD 50 11 CD 53 11 F0 BF C6 28 10 D3:XXXXXXXXXX
>000210 38 AD 53 11 ED 50 11 85 28 AD 54 11 ED 51 11 AA:XXXXXXXXXX
>000220 A5 28 69 01 85 28 90 01 E8 8A 4A 66 28 4A 66 28:XXXXXXXXXX
>000230 4A 66 28 C6 28 F0 35 AD 50 11 85 FA AE 51 11 86:XXXXXXXXXX
>000240 FB AC 52 11 84 FC 20 7D 08 A0 12 A5 FD 20 00 08:XXXXXXXXXX
>000250 C8 A5 FC 20 00 08 A9 FF A0 1F 20 00 08 18 A5 FC:XXXXXXXXXX
>000260 69 01 85 FC 90 02 E6 FD C6 28 10 D0 AD 53 11 29:XXXXXXXXXX
>000270 07 AA C9 07 F0 23 A9 00 38 6A CA 10 FB 85 28 AD:XXXXXXXXXX
>000280 53 11 85 FA AD 54 11 85 FB AC 52 11 84 FC 20 7D:XXXXXXXXXX
>000290 08 20 6A 08 05 28 20 C6 08 60 EA 8D 50 11 8E 51:XXXXXXXXXX
>0002A0 11 8C 52 11 AD 53 11 38 ED 52 11 85 28 E6 28 AD:XXXXXXXXXX
>0002B0 50 11 AE 51 11 AC 52 11 20 83 08 EE 52 11 C6 28:XXXXXXXXXX
>0002C0 D0 ED 60 EA 0A 0A 0A 0A 86 FB 05 FB A0 1A 20 00:XXXXXXXXXX
>0002D0 08 60 EA EA 00 00 00 00 00 00 00 00 00 00 00:XXXXXXXXXX
LISTING 4

```

de MicroDrukker

Textshop/ desktop publishing center

Roelof Hartstraat 27, Amsterdam 020-644659

Bespaar honderden guldens op zetkosten voor briefpapier, facturen etc.

Pluk de vruchten van de DTP techniek, bestel nu een complete set originelen voor uw zakelijk briefpapier, vervolgblad, factuur, aanmaning en declaratie-formulier. De complete set originelen, die u zo naar de (snel)-drukker of copyshop kunt brengen, kost u slechts f 100,- incl BTW.

Bel op voor een voorbeeld, geef dan uw gegevens op, de hele set komt per post bij u binnen.

Het wissen van Basicprogramma's uit het Basicgeheugen kan op zeer eenvoudige wijze met het Reset-commando. Het is veel sneller en prettiger dan het telkens uitzetten van de computer. S. Hol behandelt hieronder de verschillende mogelijkheden om bij de Commodore 64 met dit commando te kunnen werken.

Wissen van het Basicgeheugen

Reset

De reset, of wel het resetten is een (snelle) manier om in het basicgeheugen de basicprogramma's te wissen. Ook stopt een machinetaal-routine door een reset. Het bewijs vinden we in de praktijk. Laad maar eens een in machine-code geschreven programma, en list dit na het laden. Ont-houd het begin-adres (de verschenen SYS met een decimale waarde). Run het programma en reset het. Tik dan de SYS in met het beginadres, het programma zal in 90% van de gevallen weer lopen.

Waarom Reset

Vaak is de enige mogelijkheid om weer ruimte in het geheugen te krijgen, naast resetten de computer uitzetten, (soms kan het ook met RUN/STOP + RESTORE, maar dat is maar een enkele keer). Maar ook wanneer storende programma's verwijderd moeten worden zit er vaak niets anders op, dan de machine uit en aan te zetten.

Dat uitzetten van de computer is niet goed voor het apparaat, vandaar het nut van een reset-commando. Reset heeft nog een groot voordeel: reset werkt namelijk toch nog als we een programma niet d.m.v RUN/STOP + RESTORE kunnen stoppen, wat vaak het geval is.

Terughalen na een reset

Als men eenmaal een programma heeft ge-reset, is het basicgedeelte van het programma verdwenen. Dit is makkelijk terug te halen door de volgende opdracht:

```
POKE 2050,8:SYS42291:
POKE 46,
PEEK(35)-(PEEK)???253):
POKE 45,
PEEK(781)+2AND255
```

Dus als u een (basic)programma na een 'onbedoelde' reset wilt her-listen, -saven, en runnen, type dan de bovenstaande regel in.

Hoe werkt Reset

Het resetten van de computer kan op verschillende manieren. We onderscheiden naast het uitzetten vier manieren:

- ° d.m.v. de serial I/O-poort
- ° d.m.v. de user I/O-poort
- ° d.m.v. een eventuele cartridge in de cartridgepoort (iets wat we hier niet behandelen)
- ° d.m.v. een basic commando (het straten van een machinetaalroutine)

Serial I/O

Als we de seriële poort (aan de achterkant de derde ingang van rechts) bekijken ontdekken we zes connectoren. Om nu te resetten gebruiken we 'ingang' twee en zes en 6. De twee staat voor ground (GND), en de zes voor reset (RST).

Door deze twee te verbinden reset men de computer. De schakeling gaat als volgt:

Benodigheden:

- ° 1x passende plug
- ° 1x drukschakelaar, die na het drukken geen contact meer maakt.
- ° 1x 30 cm lange geïsoleerde koperdraad.

User I/O

De userport (achterkant: de meest rechtse) heeft 24 ingangen en is als reset-functie de enige oplossing naast

de cartridge-reset als u een drive heeft, want deze moet op de seriële poort worden aangesloten. Van achter gezien ziet de userport er zo uit:

Reset is nu mogelijk door 'ingang' een (ground:GND) en drie (reset:RST) te verbinden. De schakeling ziet er voor de userport zo uit:

Benodigheden:

- ° 1x passende plug op userport
- ° 1x zelfde schakelaar als bij serial port
- ° 1x 30cm lange geïsoleerde koperdraad.

SYS-commando

Er is nog een vierde manier en dat is een SYS-commando, maar daar hebben we niets aan als we een programma draaiend hebben (we kunnen dan namelijk geen opdrachten geven). Toch is het commando handig dus hier is het:

SYS 64738 (\$FCE2)

Tot zover de mogelijkheden om uw Commodore 64 te resetten. veel succes!

S. Hol



Luc Sala's Datakolom

Terminologie

Het zijn in de computerwereld niet alleen de machines en de software die veranderen, ook het computer-jargon ontwikkeld zich. In het Nederlands hebben we daar niet zo'n last van, het zijn vooral de Engelse termen, die steeds fraaier worden. Dat komt voor een deel omdat bedrijven nieuwe termen moeten verzinnen voor functies of onderdelen.

Een goed voorbeeld daarvan is Microsoft, dat het begrip Treads (draden?) invoerde met OS/2. Maar dat is nog te vergeven. Het zoeken naar woorden om iets heel nieuws duidelijk te maken is nodig en soms komen er ook best aardige vondsten uit. Maar wanneer IBM niet meer zegt, dat de oudere modellen niet meer leverbaar zijn, maar spreekt over "functionally stabilized", dan wordt het lachwekkend. Want de oude AT en XT modellen zijn gewoon niet meer te koop en men ontwikkelt er dus geen nieuwe dingen meer voor. Waarom is het dan nodig om dat allemaal schitterend te verhullen in onbegrijpelijke en misleidende termen.

Apple doet weer op een andere manier aan taalvervuiling. Daar heeft men enerzijds de neiging om gewild creatief te willen zijn, met bommetjes op het scherm in plaats van een ook

begrijpelijk "error" boodschap. Maar men gaat nog een stapje verder. In de pogingen de Macintosh te verheffen tot een soort half-goddelijke status, past men ook het vocabulaire aan. In plaats van gewoon termen als **get** of **open** te gebruiken, spreekt men nu over "invoke", en dat heeft een soort mystieke klank, helemaal passend bij de paternalistisch neerbuigendheid naar de computer-gebruiker, door het bedrijf uit Cupertino. De echte Mac-gelovigen vinden dat prachtig, zij willen namelijk geen werktuig, maar een afgodsbeeld hebben, dat bovendien nog doet wat zij willen. Zich onderwerpen aan de beeldtirannie der iconen, en ademloos luisteren naar wat de Apple hogepriesters verkondigen, hoort daar helemaal bij.

Apple, een bedrijf dat mede groot geworden is door de "Hype", de opwindende in de media rond nieuwe producten, heeft zich daar nu ook publiekelijk

toe bekeerd, alleen hebben ze een R achter het woord gezet en nu is het nieuwe buzz-woord Hyper geworden. Een laatste voorbeeld van verwrongen taalgebruik is "platform". Op allerlei conferenties en tentoonstellingen praat men niet meer over een bepaalde computer of een operating systeem, maar over een platform. Er is een Unix platform, een Mac-platform, een 386 platform, het klinkt allemaal veel interessanter dan dat je op een micro werkt. Maar we zullen er wel weer aan wennen, zolang we ons gezond verstand-platform maar geregeld "invoken".

Luc Sala.

Op uw C64/C128 of MS DOS machine op **EENVOUDIGE** wijze uw **KAS, BANK en GIRO** bijhouden met:

**Beurs 64 of
PC Beurs**

f 99,00
(incl. BTW)
franco

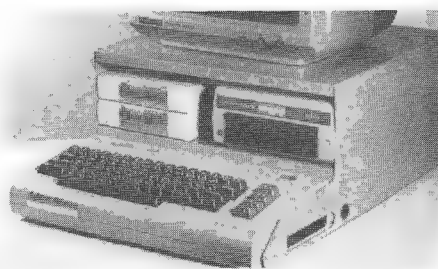
Beurs 64 is een menugestuurd programma.

Zie: Commodore Info 1/87

MARVELD COMPUTERFRAMES

Voor C-64 en
MSX computers

f 169,-
Incl. btw
franco



MARVELD printersteunen
Voor de prijs hoeft
u het niet te laten:
slechts **f 29,95 incl. BTW.**
(excl. verzendkosten)

COMPUTERTAFELS

BONDWELL PC'S

3 jaar GARANTIE

MarvelD Computing, Berkenlaan 57, Malden. Tel. 080-583522 of 08851-17583

In HiFi-winkels en demonstratieruimten wordt vaak zeer veel geschakeld om de verschillende apparaten in diverse combinaties te laten horen. De bronnen kunnen gekozen worden: CD-spelers, kassettedecks, platenspelers, tuners; de versterkers kunnen geschakeld worden: merk A, merk B enz. Kees van der Vlies belicht een marktsegment waarin de computer goede diensten kan bewijzen.

computersturing en hifi

HIFI-SCHAKEL-UNITS

Het is voor de klant in de meeste HiFi-zaken scherp opletten en goed onthouden wat er nu precies gedemonstreerd wordt: Was die versterker nr.1 nu een ONKYO of een LUXMAN? Kostte die BOSE-luidspreker nu f 395 of f 595? Luister ik nu naar het cassettedeck of de radio?

De verkoper moet met een liefst eindeloos geduld talloze malen schakelen van de ene combinatie naar de andere. De klant kan zich na verloop van tijd wat ongemakkelijk gaan voelen als hij erg lang beslag legt op de bereidwilligheid van de verkoper.

Hier lijkt een taak voor de computer weggelegd. Het geduld van de computer is eindeloos, de schakelmogelijkheden zijn zeer talrijk en de bijbehorende informatie kan via een beeldscherm zichtbaar gemaakt worden. Toch heeft het tot nu toe ontbroken aan gebruikersvriendelijke schakel-eenheden. Dit heeft ermee te maken dat het schakelen van hifi-komponenten iets anders is dan het schakelen van de huiskamerverlichting of het koffiezetapparaat. Het gaat namelijk om zeer complexe en vrij zwakke signalen, die gevoelig zijn voor storingen: schakelklikken, brom, ruis, vervorming, niveauverschillen, aantasting van het frekwentiebereik etc.

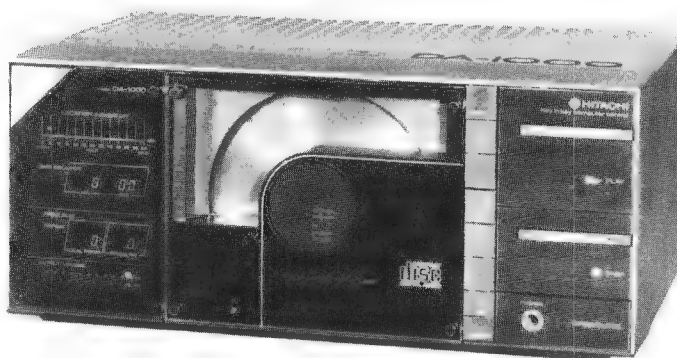
Een Nederlands produkt

De Nederlandse firma Lumibel Software Enterprises in Arnhem heeft een complete set op de markt gebracht, waarmee hifi-apparatuur op alle mogelijke manieren automatisch geschakeld kan worden. De kastjes heten: Robot 101, Robot 202 en Robot 303. Alleen de computer en de vereiste kabels behoeven eraan gekoppeld te worden en de installatie draait. De meegeleverde software is bestemd voor de Commodore-64 of

128, maar ook een MSX-versie is al beschikbaar en andere versies (PC?) zijn of worden ontwikkeld.

° 7) beveiliging van de aangesloten apparatuur tegen diefstal

De ingevoerde gegevens kunnen te allen tijde door de gebruiker gewijzigd of aangepast worden. Men heeft zich dus qua veelzijdigheid geen beperkingen willen opleggen.



De apparatuur

De kasten zijn in de bekende 19-inch racks te monteren, maar kunnen ook los staan.

De Robot 101 is bestemd voor het schakelen van maximaal 15 luidspre-

Tal van zaken zijn hiermee te programmeren:

- ° 1) de met elkaar te verbinden componenten
- ° 2) de volgorde waarin de onderscheiden combinaties 'automatisch' klinken
- ° 3) de speelduur van iedere combinatie
- ° 4) de op het beeldscherm weer te geven informatie
- ° 5) overslaan van minder gewaardeerde combinaties
- ° 6) lichtsignalering van de ingeschakelde componenten

kerparen (waarvan eventueel 5 stel actieve boxen).

De Robot 202 kan verschillende versterkers (maximaal 15) schakelen en heeft, zoals de andere Robots, een uitgang voor directe LED-aansturing, welke m.b.v. relais ook voor aansturing van spots te gebruiken is.

De Robot 303 schakelt (maximaal 15) bronnen en bewaakt de aangesloten apparaten. Daarvoor is zelfs een accu ingebouwd. Geen der aangesloten apparaten kan ongemerkt weggehaald worden. Het uitgangssignaal kan ook voor opname op een recorder gebruikt worden (3 uitgangen).

In (lijn)niveau-aanpassingen is voorzien, zodat niet telkens de volumeregelaar hoeft te worden bijgesteld. Voor handbediening zonder computer is een los toetsenbord leverbaar. Uiteraard vervalt dan de programmeermogelijkheid.

Alle 'Robots' kunnen dubbel worden uitgevoerd, zodat de capaciteit vergroot wordt; ook samenvoegen van verschillende units is mogelijk. In dergelijke gevallen zal de programmatuur aangepast worden.

De software

De verschillende schakelfuncties kunnen gemakkelijk ingevoerd worden; ook de bij elke functie behorende tekst, die op het scherm moet verschijnen. De gebruiker kan deze zelf wijzigen. Overigens is de toegang tot het programma wel via een code-woord beveiligd, zodat de klant of een andere onbevoegde niet in het programma kan gaan pionieren.

Via een vraag- en antwoordprocedure geeft de gebruiker aan welke functies en welke schermteksten (bv. menu's)

geprogrammeerd worden. In eerste instantie zullen dat de aangesloten componenten zijn, waaruit gekozen zal worden, dan de volgorde waarin ze te horen moeten zijn, automatisch overschakelen met tijdsduur per combinatie of handmatig overschakelen, invoer van ongewenste componenten (die dan overgeslagen zullen worden), namen, prijzen en gegevens van de afzonderlijke componenten of combinaties en eventuele andere vermeldenswaardige zaken. Uiteraard kunnen alle ingevoerde gegevens op diskette worden opgeslagen.

Kortom allemaal aan de praktijk ontleende toepassingen.

Hardware

De kasten zijn duidelijk voor professionele toepassingen ontworpen. De behuizing is degelijk, de connectoren zijn 'gold plated' en - dit is wel het belangrijkste - de relais zijn degelijk uitgevoerd met goudcontacten. Pure Hi-Fi-fanaten kunnen namelijk subtiele klankveranderingen waarnemen, die veroorzaakt worden door (mechani-

sche) schakelaars in de signaalweg. Dit is vaak ook een kwestie van overtuiging bij de betreffende audiofielen; die door de solide uitvoering van de schakelkasten wel niet aan het wankelen gebracht zal worden. Wij hebben geen hoorbare klankaantasting kunnen waarnemen. Opvallend was het ontbreken van schakelklikken en de naadloze overgangen, een bewijs van vakkundige uitvoering. De computer wordt met een vlakbandkabel aan de Robots gekoppeld. Bij de luidsprekeraansluitingen kunnen de dikke speciale speakerkabels worden toegepast, zodat men niet beperkt is tot het schemerlampsnoer. Het beeldscherm moet natuurlijk op een plaats opgesteld worden waar de klant er een goed zicht op heeft. Zonodig kan een beeldsignaal-versterker tussen geschakeld worden.

De Robot 101 en 202 kosten ieder f 1.970,-. Het programma gaat f 640,- kosten, de flatcable (10 m) f 125,- en het losse toetsenbord (alleen nodig bij gebruik zonder computer) f 100,-. Dit zijn prijzen ex. BTW. Inlichtingen: tel. 085 - 432446

Nieuws uit de Demowereld

Naast de gebruikelijke demo's die uit hobby-overwegingen worden geproduceerd staan de computerkrakers ook garant voor een aantal gebruikersprogramma's. Vaak worden deze programma's gemaakt ter ondersteuning van het vele programmeerwerk.

Zo bestaan er demo-editors die qua gebruikersvriendelijkheid niet onderdoen voor de commerciële softwarepakketten. Met deze editors kunnen complete demo's worden gemaakt zonder dat men iets af hoeft te weten van programmeren.

Crunchers

Een ander veelgebruikt programma is de zogenaamde 'cruncher'. Deze 'crunchers' verkleinen programma's zodat ze minder ruimte innemen op diskette of cassette. Het principe van deze 'crunchers' lijkt op het vacuum trekken van een plastic zakje. De 'cruncher' verwijdert alle nullen uit het programma en zet de overgebleven getallen zo dicht mogelijk bij elkaar. De bekendste 'crunchers' zijn: the Timcruncher van **The Mercenary Cracker** en de 1001 Cardcruncher die zelfs op eeprom(!) geleverd wordt.

Verder wordt de commerciële software ook aangepast door krakers en de-

momakers. Zo is er door the **New Generation Crew** een 'super-versie' gemaakt van het pakket Super-Script. Een groep die ook een programma aanpaste is the **Dutch Usa-team**. Zij veranderden het programma 'Sound monitor' in een 'Rockmonitor'. De 'sound-monitor' waarmee eerst alleen 3-stemmige muziek gemaakt kon worden kreeg een vierde stem en wel in de vorm van gedigitaliseerde drumgeluiden. Als de gebruiker de beschikking heeft over een digitizer kunnen zelfs eigengemaakte geluiden in de muziek verwerkt worden.

Dansende sprites

Een groep die vooral gebruik maakt van 'rock-monitor' is **The Fashion Crew**. Van hun hand kwam de bekende demo 'Mighty Sprites'. Hierin laten zij op zelfgemaakte muziek een aantal sprites op het beeldscherm 'dansen'. De bewe gingen van deze sprites kunnen door de gebruiker zelf veranderd worden door het indrukken

van een aantal toetsen. Het aantal mogelijke bewegingen betreft hier $255 \times 255 = 65535$ (!)

Ook een prachtige serie van demo's hebben **The Judges** gemaakt. Zij zijn de bedenkers van een zeer opmerkelijke routine, die zij *Flexible Line Distance* genoemd hebben. Hiermee kunnen wonderlijke effecten bereikt worden, door bijvoorbeeld delen van een high-resolution tekening onafhankelijk van elkaar te laten schuiven.

Maar ondanks al deze nieuwe routines die in 1987 zijn gemaakt is duidelijk merkbaar dat de Amiga de C64 gaat vervangen. Deze overgang zal natuurlijk nog wel wat tijd vergen dus hoop ik jullie in 1988 nog veel nieuws uit de fascinerende wereld der demo's te brengen.

Darius Heydarpour

Vragen van gebruikers

Printerbersturing

De heer L. Vandenbosch in Diepenbeek vraagt naar de mogelijkheden om de kop van de printer via software te kunnen besturen. Hij heeft zelf het één en ander zonder succes geprobeerd.

De oplossing van dit probleem kun je al vinden in ieder Commodore Basic boek, maar de ervaring leert, dat het betreffende commando meestal wordt genegeerd. De opdracht in kwestie is de SPC()-functie.

We kennen allemaal inmiddels wel de manier om tekst op het scherm te zetten, ook met hulp van de Basic-functie TAB(). Welnu, SPC() is bijna identiek aan TAB(), met dit verschil, dat de parameter van SPC() gebruikt wordt om te tellen vanaf de huidige positie van printkop, cursor of lees/schrijfkop, terwijl TAB() als functie telt vanaf de meest linkse positie op het scherm. Als voorbeeld de volgende listing:

```
10 REM Verschil SPC en TAB
   functie
20 FOR I = 1 TO 20
30 PRINT TAB(I); "Commodore";
40 NEXT I
50 PRINT:PRINT:
60 FOR I = 1 TO 20
70 PRINT SPC(I); "COMMODORE";
80 NEXT I
```

De functie SPC() wijkt duidelijk af van de TAB() en wordt, wegens onbekendheid, daarom ook tamelijk weinig toegepast. In feite is dat jammer, want TAB() is alleen te gebruiken voor schermoutput, terwijl SPC() ook kan worden aangewend voor andere devices, niet alleen printers, maar ook modems, taperecorders en diskdrives.

Voor de volledigheid nog even wat meer over SPC(). De parameter die aan deze functie kan worden meegegeven moet liggen tussen 0 en 255. Diskfiles kunnen slecht tot 254 gebruikt worden, omdat de offset in één diskblock al twee bytes nodig heeft voor interne pointeropslag.

Programma algoritmes

Het gebeurt nogal eens, dat we op de redactie vragen krijgen, die er op neer komen dat we met man en macht aan het programmeren moeten slaan, om één van de lezers uit z'n software-problemen te helpen. Zo ligt hier een vraag op m'n bureau om maar eventjes een complete Hi-Res softscroll routine voor de C-128 te gaan maken en te publiceren. Natuurlijk zouden we dit allemaal graag willen doen, maar we hebben er helaas de tijd niet voor en ook deze rubriek is niet in eerste instantie opgezet om voor iedere gebruiker de moeilijke klussen te gaan zitten programmeren. Aan de andere kant willen we iedereen, zover mogelijk graag helpen. Daarom hierbij een interessant verzoek, waar we in de loop de jaren gelijkluidende vragen over hebben gekregen:

Hoe kun je, in Basic, een (recursieve) routine maken, waardoor van een string alle mogelijke lettercombinaties worden uitgeprint. Dus van de string AS= "ABCD" moeten er op het scherm, of printer, alle mogelijke string-combinaties worden geprint. Dus ABCD, ABDC, ACDB, ACBD, ADBC, ADCB, BCDA, BCAD, en zo verder.

Geloof me, hoewel het uiteindelijke programma niet erg lang zal zijn, zul je er toch de nodige moeite mee hebben om een dergelijke opgave tot een goed eind te brengen. We vragen

hierbij iedereen die denkt een oplossing te hebben, z'n listing naar de redactie te sturen, dan zullen we hem publiceren. Het programma moet natuurlijk strings kunnen verwerken van iedere lengte en inhoud. Recursief betekent, dat een routine vanuit dezelfde routine weer opnieuw wordt aangeroepen.

Status-bits

Het fenomeen status ofwel de variabele ST in Basic zorgt vaak voor problemen. Zo schrijft Wim Bos uit Amsterdam, dat hij graag iets meer wil weten over ST, maar eigenlijk nergens de volledige informatie over

Beste Wim, de STatus-variabele is in feite een Integer-functie in Basic, die wordt gebruikt om bij het werken met alle I/O (input/output) aan de computer door te geven, hoe het is gesteld met het bepaalde device (printer, cassette-deck of drive) waarop een open file aanwezig is. Aan de hand van de waarde van ST kan de computer bepaalde acties nemen die door de gebruiker kunnen worden geprogrammeerd.

De STatus bestaat uit één BYTE, dus 8 bits, waarvan ieder bit zowel aan als uit kan staan. Dat betekent dus, dat we in feite slechts 8 verschillende Status-mogelijkheden hebben, of in heel bijzondere gevallen een combinatie van diverse bits, maar dit komt in de praktijk nooit voor. De waarde van de status-bit is altijd 0 (NUL) als er geen bijzonderheden zijn opgetreden tijdens de I/O van het betreffende file. Ook voor de output naar printers praten we over 'file-transfer'. Zodra de waarde van ST anders is dan NUL, betekent dit dat er iets aan de hand is. de printer kan niet aan staan of de disk is tijdens het lezen bij het EOF (end of

file) aangekomen. Door de status te testen, op een waarde van 0 of ongelijk 0 kunnen we daardoor snel bekijken of alles in orde is. We zullen daar een voorbeeld van geven, door file "TEST" van de disk te lezen:

```
10 REM LEZEN VAN DISK
100 OPEN 2,8,2,"TEST,SEQ,READ"
110 GOSUB 1000 :REM TEST
    STATUS
120 GET #2,A$
130 GOSUB 1000
140 PRINT A$;
150 GOTO 120
999 :
1000 REM STATUS TEST
1010 IF ST=0 THEN RETURN
1020 PRINT "Er is iets fout"
1030 PRINT "STATUS = ";ST
1040 CLOSE 2
1050 END
```

Tijdens het lezen van "TEST" zal alles wel goed gaan, maar probeer de disk maar eens te verwijderen, of zorg dat er geen bestand "TEST" op de disk aanwezig is. De ST zal dit onmiddellijk signaleren. Ook het EOF teken wordt door de ST-vlag gevangen, vandaar dat aan het einde van "TEST" ook de foutmelding wordt gegeven met STATUS = 64.

Enkele veel voorkomende ST waarden zijn:

```
-128 Device not present
64  END of FILE
1   Time out schrijven
2   Idem lezen
16  Mismatch of leesfout
32  Checksum fout (voor tape)
```

Het is leuk om aan ieder eigen programma een routine te koppelen, die voor iedere statusfout onmiddellijk terugkomt met één van bovenstaande meldingen.

Koppelen van Basic-listings

De familie Koffeman in Urk bezit sinds enkele weken een 128. Er wordt op een geheel eigen wijze tekst mee verwerkt, waarbij de geschreven data in feite als programma-bestand wordt behandeld, zodat op de 128 de ingebouwde help-functies zoals FIND kunnen worden gebruikt. Het probleem is nu, om ver-

Er zijn, zoals bij alle problemen, diverse oplossingen voor dit fenomeen. Hoewel ik persoonlijk van mening ben, dat het werken met

tekstbestanden bij voorkeur niet op deze wijze moet worden aangepakt, geef ik hierbij wel een manier om verschillende bestanden te kunnen MERGEN, omdat deze routine ook gebruikt kan worden voor 'echte' Basic-programma's.

De meeste MERGE-routines moeten met hulp van een klein machinetaal-programma in het computergeheugen worden gezet, waarna er pas mee kan worden gewerkt.

De 'Commodore-Info' oplossing is echter veel simpeler, en berust op het wisselen van de start- en eind-pointers van de twee Basic-programma's.

We zullen het voorbeeld vergezellen van een beetje uitleg over het reilen en zeilen binnen in de Commodore 128. Als er een programma wordt ingeladen in een 'schone' computer, dan staat zowel de pointer van begin als eind programma naar RAM adres \$0801. Zodra het programma is ingelezen, staat de eindpointer op het adres waar de laatste byte van de Basic-listing in het geheugen te vinden is. Als we nu, om het koppelen van twee programma's tot stand te brengen, de computer wijsmaken, dat het startadres niet \$801 moet zijn, maar het eind-adres van de vorige listing in RAM, dan zal de Commodore 128 zich daaraan houden, en het tweede programma netjes achter de eerste in RAM inlezen.

Vervolgens zetten we de start-pointer weer terug naar \$801, zodat er nu van twee listings één programma is gemaakt, wat daarna in z'n geheel weer kan worden weggeschreven. Nu de manier waarop we dit tot stand brengen.

Eerst moet het eerste programma op de normale manier worden geladen. Dus bijvoorbeeld :

```
LOAD "TEST1",8,1
```

Daarna typen we in direct-mode de volgende regel op het toetsenbord:

```
M = PEEK(47): N= PEEK(48) :
POKE 45,M : POKE 46, N
```

Nu laden we het tweede programma, **LOAD "TEST2",8,1**, en geven daarna de opdracht om het startadres van het complete programma weer terug te zetten:

```
POKE 45,1: POKE 46,8
```

Bij de opdracht LIST, zal nu blijken of het koppelen geslaagd is. Zorg wel, dat de twee programma's regelnummers hebben, die overeenstemmen met hun positie in RAM. Desnoods moet voor ieder programma afzonder-

lijk eerst een RENUMBER-utility worden gebruikt, zodat er geen dubbele regelnummers meer voorkomen, want daardoor kan het gebeuren, dat de computer fouten gaat maken.

Bovenstaande opdrachten zijn voor de C-128 bestemd. Wil men dezelfde bewerkingen uitvoeren op een standaard C-64, dan moeten de volgende twee regels worden ingetoetst:

```
6M = PEEK(45): N= PEEK(46) :
POKE 43,M : POKE 44, N
```

```
en : POKE 43,1: POKE 44,8
```

Kortom, een snelle en eenvoudige manier om meerdere kleine Basic-listing, of afzonderlijke subroutines te koppelen tot één groot programma.



Jan Bodzinga.

Het was weer moeilijk kiezen uit de vele ingezonden programma's. Er zaten dit keer veel mooie bij, die je dan ook allemaal hier aantreft. We proberen de programma's nu zodanig af te drukken, dat ze exact hetzelfde op papier staan, als op het scherm van je computer.

Blijf ze overigens insturen, die vondsten en ontdekkingen van jullie, naar: Postbus 112, 1260 AC, Blaricum, ten name van ondergetekende.

Basic Micro's



We zullen gelijk maar los branden met een programma dat zeker een tien met een griffel verdient. Het is afkomstig van Sam Ratajczak uit Houthalen (België).

Het tekent een prachtige drie-dimensionale grafiek op een C-64 met Simons Basic. Het programma is echter gemakkelijk geschikt te maken voor allerlei andere computers door de grafische opdrachten en eventueel wat getallen aan te passen. Door verandering van de formule voor Z=... in regel 30 en 40 kun je andere figuren te zien krijgen.

```
10 HIRES1,0:FORX=-πTOπSTEP.2:Y=-π:GOSUB3
0:FORY=-2.94TOπSTEP.2:GOSUB40:NEXTY,X
20 FORY=-πTOπSTEP.2:X=-π:GOSUB30:FORX=-2
.94TOπSTEP.2:GOSUB40:NEXTX,Y:WAIT0,0
30 Z=COS(4*SQR(X*X+Y*Y))/6:AA=139+28*(Y-
X/2):BB=95-54*(Z-X/2):RETURN
40 Z=COS(4*SQR(X*X+Y*Y))/6:A=139+28*(Y-X
/2):B=95-54*(Z-X/2)
50 LINEAA,BB,A,B,1:AA=A:BB=B:RETURN
```

Versleten toets

Een handige utility voor mensen met een versleten C-64 levert T. Bressers uit Bergeyk. Je kunt namelijk een niet (goed) meer werkende toets "vervangen" door een andere. De gebruiksaanwijzing is als volgt: je typt de volgende Micro in, en je RUNt hem. Laad nu een ander programma of typ het in, waarbij je op de "POND"-toets drukt, in plaats van op de versleten toets. Als je dan **POKE 49168,ASCIIcode:SYS49152** tikt, en dan **LIST**, dan zal het "POND"-teken vervangen worden door het karakter met de betreffende ASCII-code. Bedenk wel dat het zo nog niet in het geheugen zit, dus daarna ga je de betreffende regels opnieuw met RETURN in het geheugen stoppen. We hebben het programma aangepast voor de Power Cartridge, dus dat is ook in orde. Pas op dat je het programma niet RUNt nadat je de SYS al hebt ingetikt, want dan zorgen de eerste twee POKE-opdrachten in regel 10 ervoor, dat elke volgende LIST-opdracht meteen blokkeert. De NEW zit er dan ook in om dat te voorkomen.

```
10 POKE49178,PEEK(774):POKE49179,PEEK(77
5):FORI=49152TO49177:READQ:A=A+Q
20 POKEI,Q:NEXT:IFA<>2550THENPRINT"FOU
T IN DATA!":END:DATA169,11,141,6,3,169
30 NEW:DATA192,141,7,3,96,201,92,208,8,1
69,45,32,71,171,76,0,167,41,255,76
```

Cirkelstraal

Voor de C-128 is de volgende Micro. Hij is ingestuurd door Pierre Desmet uit Wervik (België). Dit programma tekent vanuit een punt een aantal lijnen van willekeurige lengte in de richting van een denkbeeldige cirkel.

```
10 PRINTCHR$(147):COLOR4,1:COLOR0,1:GRAP
HIC1,1:COLOR1,5:FORA=0TO2*PSTEP.03
20 X=INT(RND(0)*90+10):DRAW1,160,100TO16
0+X*COS(A),100+X*SIN(A):NEXT
```

En daar kun je ook nog het volgende aan toevoegen, voor een leuk effect op het eind:

```
30 SLEEP5:GRAPHIC3,0
```

'Screen Effect'

Lucien Hiens (12 jaar) uit Wassenaar stuurde een Micro in, en hij zorgt daarmee voor een kleurig 'Screen Effect' op de C-64:

```
10 FORI=49152TO49194:READA:POKEI,A:X=X+A
:NEXT:IFX=4633THENSYS49152:DATA140,32
20 PRINT"FOU!":DATA208,140,33,208,162,0
,232,224,110,240,3,76,8,192,192,2,240
30 DATA8,192,1,240,9,192,6,240,10,160,1,
76,0,192,160,6,76,0,192,160,2,76,0,192
```

Spiraal

Alweer uit België is de volgende Micro. De auteur is Ronny Vandevenne uit Berchem (Antwerpen). Het is de bedoeling dat deze Micro draait op de C-64 met Simons Basic, maar het kan weer gemakkelijk aangepast worden voor andere computers.

Het programma tekent een soort melkwegstelsel of spiraalnevel op het scherm. Door de stapgrootte in regel 10 te variëren, ontstaat een grote verscheidenheid aan figuren.

```
10 R=5:COLOUR0,0:HIRES1,0:FORA=1TO560STEP
P.5:X=R*COS(A)+160:Y=R*SIN(A)+100
20 PLOTX,Y,1:R=R+.5:NEXT:TEXT40,4,"MILKY
WAY",1,4,30:POKE198,0:WAIT198,1
```

Lichtkrant

Obbe Vermey uit Cappelle aan de IJssel stuurde weer enkele programma's in voor de Commodore 64. De nu volgende is eigenlijk meer een Basic MACRO dan een Micro, maar toch plaatsen we hem. Het programma verzorgt een mini-lichtkrant in een sprite, met drie karakters naast elkaar, en bijna drie boven elkaar, waarbij de letters nu eens naar boven wegschuiven.

Helaas werkt dit programma niet goed met de Power Cartridge, dus als je die hebt, moet je eerst QUIT intikken. Voor dit probleem is overigens wel een oplossing te vinden, maar dan moet bijna het hele programma in machinetaal omgezet worden.

```
10 FORV=0TO46:READA:POKE49152+V,A:Q=Q+A:
NEXT:K(1)=15:K(2)=22:K(3)=5
20 IFQ7111THENPRINT"FOOT IN DATA":END
30 V=53248:POKEV+21,1:POKEV,148:POKEV+1,
60:POKEV+32,0:POKEV+33,0:POKEV+39,1
40 POKEV+29,1:POKE2040,128:POKEV+21,1:FO
RT=8192TO8257:POKET,0:NEXT
50 FORT=0TO7:SYS49152:POKE808,225:POKE56
334,PEEK(56334)AND254
60 POKE1,PEEK(1)AND251:FORG=1TO3:POKE825
4+G,PEEK(K(G)*8+V+T):NEXT
70 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334
)OR1:POKE808,237:NEXT
80 READK(1),K(2),K(3):ON-(K(1)-1)GOTO50
:RESTORE:FORG=0TO46:READA:NEXT:GOTO80
90 DATA169,32,133,254,169,63,133,253,160
,0,177,253,133,2,198,253,198,253,198
100 DATA253,177,253,170,165,2,145,253,13
4,2,166,253,16,237,232,240,10,24,165
110 DATA253,105,67,133,253,76,8,192,96
120 DATA32,32,32,4,9,20,9,19,32,8,5,20,4
,1,14,32,32,32,32,32,32,32,32,-1,,
```

In regel 120 en eventueel verder moet je de tekst in DATA-regels zetten, met de schermcodes zoals die te vinden zijn in appendix B van de Reference Guide en in appendix E van de Engelstalige handleiding.

Vermoeiend

Van dezelfde auteur komt ook nog een snelheids-spelletje: geef een flinke roffel op de spatiebalk, en deze Micro

vertelt je hoe vaak je de schakelaar hebt ingedrukt. Het record staat op 881, wie verbetert dit?

```
10 PRINTCHR$(147):POKE650,64:A=0:FORT=0T
O450:GETA$:IFA$=CHR$(32)THENA=A+10
20 A=ABS(A-1):PRINTCHR$(145):NEXT:POKE6
50,0
```

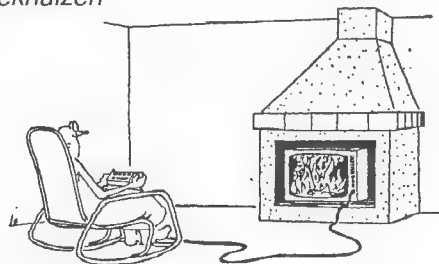
Grotere letters op het scherm

Als klap op de vuurpijl volgt hier nu nog een MACRO die - net als de mini-lichtkrant overigens - zorgt voor dubbel-hoge letters op het scherm, maar ditmaal zonder sprites. Het is afkomstig van Ron en Kim Brobbel uit Vlaardingen. De grotere karakters worden verkregen door de helft van de bestaande karakterset dubbel te kopiëren naar het geheugen, en daarna twee karakters te gebruiken voor elk nieuw teken, namelijk het karakter zelf, en het inverse karakter (dat dan niet invers is, maar de onderste helft). Klinkt ingewikkeld, maar probeer het maar eens, een demo is ingebouwd. Je kunt ook de laatste A en B in regel 30 veranderen in nullen, dan krijg je letters die uit strepen zijn opgebouwd.

Helaas geldt ook hier dat je bij een Power Cartridge eerst QUIT moet intikken, tenzij je de eerste regels omzet in machinetaal.

```
10 PRINTCHR$(147):V=53248:POKEV+32,0:POK
EV+33,0:POKEV+24,31:POKE56334,0
20 POKE1,51:FORT=0TO1023STEP8:FORG=0TO7S
TEP2:A=PEEK(V+T+G/2):POKE14336+T+G,A
30 B=PEEK(V+4+T+G/2):POKE15360+T+G,B:POK
E14337+T+G,A:POKE15361+T+G,B:NEXTG,T
40 POKE1,55:POKE56334,1:FORK=0TO5:READA$
,V:L=LEN(A$):IFLAND1THENA$=A$+" ":L=L+1
50 FORA=1TOL/2:B$=LEFT$(A$,A)+RIGHT$(A$,
A):H=20-LEN(B$)/2:POKE214,V:SYS58640
60 PRINTTAB(H)CHR$(158)B$:PRINTTAB(H)CHR
$(129)CHR$(18)B$:FORR=0TO30:NEXTR,A,K
70 S=54272:FORT=0TO23:POKES+T,0:NEXT
80 POKES+24,15:POKES+5,10:POKES+6,254:PO
KES,21:POKES+2,20:POKES+3,1:POKES+4,65
90 POKES+1,1+10*RND(1):FORT=0TO100:NEXT:
POKES+1,10:GOTO90
100 DATA*** DIT IS EEN PROGRAMMA INTRO *
**,3,GEMAAKT VOOR COMMODORE INFO 1987,5
110 DATABELANGRIJK,9,DE STRING MAG NIET
LANGER ZIJN,11
120 DATADAN 38 KARAKTERS,13,** RAK SOFTW
ARE 1987 **,20
```

Peter Broekhuizen



De negende aflevering van onze cursus 65XX machinetaal wordt voor het overgrote deel in beslag genomen door de allang beloofde listing van een eigen, echte mini-assembler. Dit programma zal in de komende afleveringen zijn nut gaan bewijzen, omdat de toekomstige programma-voorbeelden in machinetaal met hulp van mnemonics zullen worden gegeven. Daarom nu een aflevering specifiek over de assembler.

Cursus Machinetaal

door Tjipke van der Land

Deel 9 : De Mini-Assembler

In aansluiting op al eerder gedane beloftes zullen we deze keer de listing van ons eigen assembler/disassembler programma plaatsen. In het vervolg zullen we dit programma bijna dagelijks nodig hebben. We kunnen ons voorstellen, dat velen een zucht van verlichting slaken, bij het zien van deze listing, want we hebben er allemaal lang op moeten wachten, maar hier is hij dan.

De gepubliceerde listing werkt in principe net als alle andere (mini) assemblers die bestemd zijn voor de Commodore 65XX processor. Beschik je inmiddels reeds over een eigen assembler, dan adviseer ik, om de opdrachten van deze kleine, maar veelomvattende assembler eens te bekijken, en als de mogelijkheden min of meer overeen komen, kun je bij het volgen van de lessen je eigen assembler gebruiken, zodat je het intypen van dit programma besparen kunt. Overigens zal deze assembler ook verschijnen als Infolist-diskette, die via de gangbare kanalen in je bezit kan komen.

Zure appel

Vanwege het gebrek aan ruimte kunnen we in deze aflevering niet de volledige beschrijving van alle mogelijkheden plaatsen, maar buiten dat om heb je uren van intoetswerk voor de boeg, en zul je daarna wellicht niet meer de lust hebben om eens flink in machinetaal te gaan programmeren. We geven natuurlijk wel wat voorbeelden waarmee de assembler aan het werk kan worden gezet.

Hieronder staat de complete listing van de assembler, in de vorm van een Basic programma met nogal wat DATA-regels, die zo onder de hand wel bekend zijn.

Even door de zure appel heenbijten en het hele programma voor later netjes wegschrijven op tape of disk, waardoor je in het vervolg altijd een as-

sembler voor je Commodore bij de hand hebt.

Het grootste probleem waar je mee te maken zult krijgen, is dat je verkeerde data intypt. Dit wordt tijdens het runnen wel gecontroleerd door een zgn. checksum algoritme, die je verteld of je data wel of niet goed ingevoerd hebt. Is het niet goed, dan is het jammer, want dan moet je de hele listing gaan controleren. Krijg je echter de melding 'Data goed ingevoerd', dan kun je met de assembler aan de slag.



Werking

Om je even iets op gang te helpen met het werken met de assembler, gebruiken we een oud voorbeeld om even te kijken wat je allemaal kunt doen met dit programma. We gebruiken daarvoor het onderstaande Basicprogrammaatje om een naam in het RAM-geheugen te zetten.

Naam: J A N K L A A S S E N

**ASCII: 4A 41 4E 20 4B 4C 41 41 53
53 45 4E**

**Decimaal: 74 65 78 32 75 76 65 65
83 83 69 78**

**10 REM poke programma voor
machinetaal**

15 REM zie MT aflevering 8

**20 S = 8192 : REM START ADRES
NAAM**

30 READ A

40 IF A = -1 THEN END

50 POKE S,A

60 S = S + 1

70 GOTO 30

**100 DATA
74,65,78,32,75,76,65,65,83,83,6
9,78,-1**

Als je dit programma één keer gerund hebt staat de naam 'JAN KLAASSEN' al in het RAM-geheugen.

Voor het invoeren van het onderstaande programma, moet je eerst de assembler hebben geladen en gerund, waarna je alleen maar hoeft in te typen (na het starten van de assembler):

. A 1000 LDA #17

denk daarbij om de eerste PUNT (.)
Je zult zien dat de assembler automatisch terug komt met het volgende adres (\$1002) waardoor je alleen maar de instructies hoeft in te toetsen zoals ze hierna staan afgedrukt:

Voorbeeldprogramma:

MNEMONIC HEX

COMMENTAAR

```
LDA #$17      A9 17      ;Getal laden ASCII laag
STA $D018     8D 18 D0    ;Parameter plaatsen
LDA $2000     AD 00 20    ;Ophalen letter 1
STA $0578     8D 78 05    ;Plaats letter in videogeh.
LDA $2001     AD 01 20    ;Ophalen letter 2
STA $0579     8D 79 05    ;Plaats letter in videogeh.
LDA $2002     AD 02 20    ;Enz.
STA $057A     8D 7A 05
LDA $2003     AD 03 20    ;Ophalen van spatie
STA $057B     8D 7B 05    ;Plaats spatie in videogeh.
LDA $2004     AD 04 20    ;Enz.
STA $057C     8D 7C 05
LDA $2005     AD 05 20
STA $057D     8D 7D 05
LDA $2006     AD 06 20
STA $057E     8D 7E 05
LDA $2007     AD 07 20
STA $057F     8D 7F 05
LDA $2008     AD 08 20
STA $0580     8D 80 05
LDA $2009     AD 09 20
STA $0581     8D 81 05
LDA $200A     AD 0A 20
STA $0582     8D 82 05
LDA $200B     AD 0B 20
STA $0583     8D 83 05
NOP           EA
BNE #$FC      D0 FC      ;LOOP VIA NOP (NO OPERATION)
```



Na het invoeren van dit programma kun je het resultaat nog eens overzien door in te tikken: **. d 1000**.

Hier zie je dan het **gedisassembleerde** programma wat je zelf hebt ingevoerd. Om het te runnen moet je invoeren: **. g 1000**.

Nu zie je de naam op het scherm verschijnen, net als de vorige keer in Basic, alleen heb je er meer moeite voor moeten doen. Je kunt het programma afbreken door te drukken op de run/stop restore. Wat problemen kan geven is de instructie **BNE #\$FC**. Deze kun je vervangen door **JMP \$ 1000**, en dan zal het allemaal zeker werken.

Met de assembler kunnen we ook programma's opslaan en weer laden, printen enz. Al dit soort zaken zullen we de volgende keren wat uitvoeriger behandelen, om optimaal met de assembler te kunnen werken. Waar het nu om gaat is dat je de assembler werkend krijgt, zodat we er de volgende keer probleemloos mee aan de slag kunnen. Vind je het invoeren te veel werk, dan kan je, zoals gezegd, de assembler bij ons bestellen via Infolist, Huizen. Als je echter zelf aan de slag gaat - een nuttige typeoefening overigens - dan wens ik je erg veel sterkte en succes bij het opsporen van de fouten.

Assembler listing Commodore 64

```
10 REM *****
20 REM *   COMMODORE INFO   *
30 REM *   MINI ASSEMBLER   *
40 REM *
50 REM * PROGRAMMA INVOEREN, EN TYP *
60 REM * DAARNA RUN (RETURN). *
70 REM * TYP NA HET RUNNEN SYS 49152 *
80 REM *
90 REM * (c) 1987 *
100 REM *****
1000 S=49152:T=0
1010 READ A:IF A=-1 THEN 1030
1020 POKE S,A:S=S+1:T=T+A:GOTO 1010
1030 IF T=283010 THEN 1060
1040 PRINT "FOUTE INVOER VAN DATA"
1050 PRINT "TUSSEN REGEL 5000 EN 6480":END
1060 PRINT "DATA IS GOED VERWERKT"
1070 PRINT "TYPE SYS 49152 VOOR HET STARTEN"
1080 END
5000 DATA 76,233,192,255,0,0,255,255,0,
0,255,255,0,0,255,255,0
5010 DATA
0,255,255,0,0,255,255,0,0,255,255,0,0
,255
5020 DATA
255,0,0,255,255,0,0,255,255,0,0,255,2
55,0
5030 DATA
0,255,255,0,0,255,255,0,0,255,255,0,
5040 DATA
255,255,0,0,255,255,0,0,255,255,0,0,2
55
5050 DATA
255,0,0,255,255,0,0,255,255,0,0,255,2
55
5060 DATA
0,0,255,255,0,0,255,255,0,0,255,255,0
,0
5070 DATA
255,255,0,0,255,255,128,0,255,255,0,0,255,255,0,0
,255
5080 DATA
255,0,0,255,255,0,0,255,255,0,0,255,255,0,0,255,2
55
5090 DATA
0,0,255,255,0,0,255,255,0,0,255,255,0,0,255,255,0
,0
5100 DATA
255,255,0,0,255,255,0,0,255,255,0,0,255,255,0,0,2
55
5110 DATA
255,0,0,255,255,0,0,255,255,0,0,255,255,0,0,255,2
55
5120 DATA
0,0,255,255,0,0,255,255,0,0,255,255,0,0,255,255,0
,0
5130 DATA
255,255,0,0,255,255,0,169,160,133,56,173,230,200
,141
5140 DATA
22,3,173,231,200,141,23,3,169,128,32,144,255,0,2
16
```

5150 DATA 104,141,62,2,104,141,61,2,104 141,60,2,104,141,59,2	5380 DATA 255,201,13,240,12,201,32,208, 209,32,121,195,144,3,32	5600 DATA 105,8,41,15,96,162,2,44,162,0, 180,193,208,8,180,194,208,2
5160 DATA 104,170,104,168,56,138,233,2, 141,58,2,152,233,0,141	5390 DATA 128,193,169,154,32,210,255,17 4,63,2,154,120,173,57,2	5610 DATA 230,38,214,194,214,193,96,32, 62,193,201,32,240,249,96
5170 DATA 57,2,186,142,63,2,32,87,198,16 2,66,169,42,32,87,195	5400 DATA 72,173,58,2,72,173,59,2,72,173 60,2,174,61,2,172,62,2	5620 DATA 169,0,141,0,1,32,204,195,32,14 3,195,32,124,195,144,9,96
5180 DATA 169,82,208,52,230,193,208,6,2 30,194,208,2,230,38,96	5410 DATA 64,169,154,32,210,255,174,63, 2,154,108,2,160,160,1,132	5630 DATA 32,62,193,32,121,195,176,222, 174,63,2,154,169,154,32
5190 DATA 32,207,255,201,13,208,248,104 104,169,154,32,210,255	5420 DATA 186,132,185,136,132,183,132,1 44,132,147,169,64,133,187	5640 DATA 210,255,169,63,32,210,255,76, 71,193,32,84,198,202,208
5200 DATA 169,0,133,38,162,13,169,46,32 87,195,169,159,32,210	5430 DATA 169,2,133,188,32,207,255,201, 32,240,249,201,13,240,56	5650 DATA 250,96,230,195,208,2,230,196, 96,162,2,181,192,72,181,39
5210 DATA 255,32,62,193,201,46,240,249, 201,32,240,245,162,14	5435 DATA 201,34,208,20,32,207,255	5660 DATA 149,192,104,149,39,202,208,24 3,96,165,195,164,196,56
5220 DATA 221,183,200,208,12,138,10,170 189,199,200,72,189,198	5440 DATA 201,34,240,16,201,13,240,41,1 45,187,230,183,200,192,16	5670 DATA 233,2,176,14,136,144,11,165,4 0,164,41,76,51,196,165,195
5230 DATA 200,72,96,202,16,236,76,237,1 95,165,193,141,58,2,165	5450 DATA 208,236,76,237,195,32,207,255 201,13,240,22,201,44,208	5680 DATA 164,196,56,229,193,133,30,152 229,194,168,5,30,96,32
5240 DATA 194,141,57,2,96,169,8,133,29,1 60,0,32,84,198,177,193	5460 DATA 220,32,136,195,41,15,240,233, 201,3,240,229,133,186,32	5690 DATA 212,195,32,105,195,32,229,195 32,12,196,32,229,195,32
5250 DATA 32,72,195,32,51,193,198,29,20 8,241,96,32,136,195,144	5470 DATA 207,255,201,13,96,108,48,3,10 8,50,3,32,150,194,208,212	5700 DATA 47,196,32,105,195,144,21,166, 38,208,100,32,40,196,144
5260 DATA 11,162,0,129,193,193,193,240, 3,76,237,195,32,51,193	5480 DATA 169,154,32,210,255,169,0,32,2 39,194,165,144,41,16,208	5710 DATA 95,161,193,129,195,32,5,196,3 2,51,193,208,235,32,40
5270 DATA 198,29,96,169,59,133,193,169, 2,133,194,169,5,96,152	5490 DATA 196,76,71,193,32,150,194,201, 44,208,186,32,121,195,32	5720 DATA 196,24,165,30,101,195,133,195 152,101,196,133,196,32,12
5280 DATA 72,32,87,198,104,162,46,76,87 195,169,154,32,210,255	5500 DATA 105,195,32,207,255,201,44,208 173,32,121,195,165,193	5730 DATA 196,166,38,208,61,161,193,129 195,32,40,196,176,52,32
5290 DATA 162,0,189,234,200,32,210,255, 232,224,22,208,245,160	5510 DATA 133,174,165,194,133,175,32,10 5,195,32,207,255,201,13	5740 DATA 184,195,32,187,195,76,125,196 32,212,195,32,105,195
5300 DATA 59,32,194,193,173,57,2,32,72,1 95,173,58,2,32,72,195	5520 DATA 208,152,169,154,32,210,255,32 242,194,76,71,193,165,194	5745 DATA 32,229,195,32,105,195
5310 DATA 32,183,193,32,141,193,240,92, 32,62,193,32,121,195,144	5530 DATA 32,72,195,165,193,72,74,74,74 74,32,96,195,170,104,41	5750 DATA 32,62,193,32,136,195,144,20,1 33,29,166,38,208,17,32
5320 DATA 51,32,105,195,32,62,193,32,12 1,195,144,40,32,105,195	5540 DATA 15,32,96,195,72,138,32,210,25 5,104,76,210,255,9,48,201	5760 DATA 47,196,144,12,165,29,129,193, 32,51,193,208,238,76,237
5330 DATA 169,154,32,210,255,32,225,255 240,60,166,38,208,56,165	5550 DATA 58,144,2,105,6,96,162,2,181,19 2,72,181,194,149,192,104	5770 DATA 195,76,71,193,32,212,195,32,1 05,195,32,229,195,32,105
5340 DATA 195,197,193,165,196,229,194,1 44,46,160,58,32,194,193	5560 DATA 149,194,202,208,243,96,32,136 195,144,2,133,194,32,136	5780 DATA 195,32,62,193,162,0,32,62,193 201,39,208,20,32,62,193
5350 DATA 32,65,195,32,139,193,240,224, 76,237,195,32,121,195,144	5570 DATA 195,144,2,133,193,96,169,0,13 3,42,32,62,193,201,32,208	5790 DATA 157,16,2,232,32,207,255,201,1 3,240,34,224,32,208,241
5360 DATA 3,32,128,193,32,183,193,208,7 32,121,195,144,235,169	5580 DATA 9,32,62,193,201,32,208,14,24,9 6,32,175,195,10,10,10,10	5800 DATA 240,28,142,0,1,32,143,195,144 198,157,16,2,232,32,207
5370 DATA 8,133,29,32,62,193,32,161,193	5590 DATA 133,42,32,62,193,32,175,195,5	5810 DATA 255,201,13,240,9,32 136,195,144,182,224,32,208,2 36

5820 DATA 134,28,169,154,32,210,255,32, 87,198,162,0,160,0,177	55,32,212,195,32,105,195	32,65,195,32,84,198,169,159,3 2,210,255,76,176,198,168
5830 DATA 193,221,16,2,208,12,200,232,2 28,28,208,243,32,65,195	6050 DATA 32,229,195,32,105,195,162,0,1 34,40,169,154,32,210,255	6280 DATA 32,191,199,208,17,152,240,14, 134,28,166,29,221,16,2,8
5840 DATA 32,84,198,32,51,193,166,38,20 8,141,32,47,196,176,221	6060 DATA 32,87,198,32,114,197,32,202,1 97,133,193,132,194,32	6290 DATA 232,134,29,166,28,40,96,201,4 8,144,3,201,71,96,56,96
5850 DATA 76,71,193,32,212,195,133,32,1 65,194,133,33,162,0,134	6070 DATA 225,255,240,5,32,47,196,176,2 33,76,71,193,32,212,195	6300 DATA 64,2,69,3,208,8,64,9,48,34,69,5 1,208,8,64,9,64,2,69
5860 DATA 40,169,147,32,210,255,169,154 32,210,255,169,22,133	6080 DATA 169,3,133,29,32,62,193,32,161 193,208,248,165,32,133	6310 DATA 51,208,8,64,9,64,2,69,179,208, 8,64,9,0,34,68,51,208
5870 DATA 29,32,106,197,32,202,197,133, 193,132,194,198,29,208	6090 DATA 193,165,33,133,194,76,70,197, 197,40,240,3,32,210,255	6320 DATA 140,68,0,17,34,68,51,208,140,6 8,154,16,34,68,51,208
5880 DATA 242,169,145,32,210,255,76,71, 193,160,44,32,194,193,32	6100 DATA 96,32,212,195,32,105,195,142, 17,2,162,3,32,204,195	6330 DATA 8,64,9,16,34,68,51,208,8,64,9,9 8,19,120,169,0,33,129
5890 DATA 84,198,32,65,195,32,84,198,16 2,0,161,193,32,217,197	6110 DATA 72,202,208,249,162,3,104,56,2 33,63,160,5,74,110,17,2	6340 DATA 130,0,0,89,77,145,146,134,74,1 33,157,44,41,44,35,40
5900 DATA 72,32,31,198,104,32,53,198,16 2,6,224,3,208,18,164,31	6120 DATA 110,16,2,136,208,246,202,208, 237,162,2,32,207,255,201	6350 DATA 36,89,0,88,36,36,0,28,138,28,3 5,93,139,27,161,157,138
5910 DATA 240,14,165,42,201,232,177,193 176,28,32,194,197,136	6130 DATA 13,240,30,201,32,240,245,32,2 08,199,176,15,32,156,195	6360 DATA 29,35,157,139,29,161,0,41,25,1 74,105,168,25,35,36,83
5920 DATA 208,242,6,42,144,14,189,42,20 0,32,165,198,189,48,200	6140 DATA 164,193,132,194,133,193,169,4 8,157,16,2,232,157,16,2	6370 DATA 27,35,36,83,25,161,0,26,91,91, 165,105,36,36,174,174
5930 DATA 240,3,32,165,198,202,208,213, 96,32,205,197,170,232	6150 DATA 232,208,219,134,40,162,0,134, 38,240,4,230,38,240,117	6380 DATA 168,173,41,0,124,0,21,156,109 156,165,105,41,83,132,19
5940 DATA 208,1,200,152,32,194,197,138, 134,28,32,72,195,166,28	6160 DATA 162,0,134,29,165,38,32,217,19 7,166,42,134,41,170,188	6390 DATA 52,17,165,105,35,160,216,98,9 0,72,38,98,148,136,84,68
5950 DATA 96,165,31,56,164,194,170,16,1 136,101,193,144,1,200,96	6170 DATA 55,200,189,119,200,32,185,199 208,227,162,6,224,3,208	6400 DATA 200,84,104,68,232,148,0,180,8 132,116,180,40,110,116
5960 DATA 168,74,144,11,74,176,23,201,3 4,240,19,41,7,9,128,74	6180 DATA 25,164,31,240,21,165,42,201,2 32,169,48,176,33,32,191	6410 DATA 244,204,74,114,242,164,138,0, 170,162,162,116,116,116
5970 DATA 170,189,217,199,176,4,74,74,7 4,74,41,15,208,4,160,128	6190 DATA 199,208,204,32,193,199,208,19 9,136,208,235,6,42,144	6420 DATA 114,68,104,178,50,178,0,34,0,2 6,26,38,38,114,114,136
5980 DATA 169,0,170,189,29,200,133,42,4 1,3,133,31,152,41,143	6200 DATA 11,188,48,200,189,42,200,32,1 85,199,208,181,202,208,209	6430 DATA 200,196,202,38,72,68,68,162,2 00,58,59,82,77,71,88,76
5990 DATA 170,152,160,3,224,138,240,11, 74,144,8,74,74,9,32,136	6210 DATA 240,10,32,184,199,208,171,32, 184,199,208,166,165,40	6440 DATA 83,84,70,72,68,80,44,65,66,194 53,194,204,193,247,193
6000 DATA 208,250,200,136,208,242,96,17 7,193,32,194,197,162,1	6220 DATA 197,29,208,160,32,105,195,164 31,240,40,165,41,201,157	6450 DATA 86,194,137,194,244,194,12,195 62,196,146,196,192,196
6010 DATA 32,254,195,196,31,200,144,241 162,3,192,4,144,242,96	6230 DATA 208,26,32,28,196,144,10,152,2 08,4,165,30,16,10,76,237	6460 DATA 56,197,91,198,138,198,172,198 70,193,255,192,237,192
6020 DATA 168,185,55,200,133,40,185,119 200,133,41,169,0,160	6240 DATA 195,200,208,250,165,30,16,246 164,31,208,3,185,194,0	6470 DATA 13,32,32,32,80,67,32,32,83,82, 32,65,67,32,88,82,32,89
6030 DATA 5,6,41,38,40,42,136,208,248,10 5,63,32,210,255,202,208	6250 DATA 145,193,136,208,248,165,38,14 5,193,32,202,197,133,193	6480 DATA 82,32,83,80,108,239,255,254,2 21,222,-1
6040 DATA 236,169,32,44,169,13,76,210,2	6260 DATA 132,194,169,154,32,210,255,16 0,65,32,194,193,32,84,198	
	6270 DATA	

Op het eerste gezicht lijkt het een vreemd idee om de Amiga als een MS-DOS machine te emuleren. De kracht van de Motorola 68000 CPU komt dan immers niet tot haar recht onder MicroSoft's disk-besturingssysteem? Voor wie echter toch MS-DOS programmatuur wil draaien is emulatie een must. Ook kan het gebruik van de Amiga als IBM PC enkele voordelen bij multitasking- en netwerktoepassingen bieden. In onderstaand artikel wordt nader ingegaan op het gebruik van de softwarematige Transformer en de hardware-emulator Sidecar voor de Amiga 500 en 1000 en de PC-kaarten voor de 2000.

Verspilling van kracht en mogelijkheden of een aanvaardbare PC- emulatie?

MS-DOS op de Amiga

Qua ontwerp zijn Commodore's Amiga's geavanceerdere machines dan een eenvoudige PC XT. De oorspronkelijke IBM gaat al zijn zevende levensjaar in en de mogelijkheden van de 8-bits (bus) Intel 8088 microprocessor zijn eigenlijk al tot op de naad uitgekleet. Daarentegen kan de 16-bits Motorola 68000 chip net als de Intel 80286 voor de AT nog wel enkele jaren mee, ondanks het feit dat de 32-bits-telgen van dezelfde families al voor de lopende band staan te dringen.

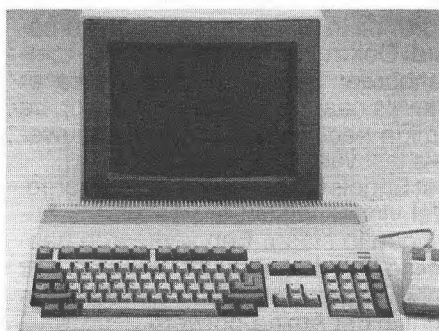
Een ander belangrijk punt is dat de geluids-, animatie- en grafische mogelijkheden van de Amiga-chips nog altijd superieur zijn aan de PC-machines in dezelfde prijsklasse. De enige argumenten om de Amiga toch als een MS-DOS computer te emuleren zijn de overvloed aan compatibele programmatuur, het gebruik van MS-DOS-netwerken en het gebruik van IBM PC hardware- uitbreidingen.

Dure randapparatuur

Ondanks het feit dat het aanbod aan Amiga software de laatste tijd sterk is toegenomen kan de hoeveelheid beschikbare programmatuur nog niet in de schaduw van de weelde der PC-pakketten staan. De keuze en variëteit aan opties is zo groot dat er voor elk wat wils te krijgen is. Bij de Amiga is er wat dezelfde branche betreft hoogstens keuze uit twee of drie programma's. Wie het gezochte niet bij de Amiga-pakketten kan vinden zal dus de beoogde software zelf moeten schrijven of de machine als een XT moeten emuleren.

Randapparatuur voor de **Amiga 500** en **1000** is momenteel nog extreem duur. Dit in tegenstelling tot harde schijven, disktestations, modemkaarten en geheugenuitbreidingen voor de IBM-PC. Daar komt nog bij dat deze beide Amiga's niet over uitbreidingsslots beschikken.

Wie een goedkope harddisk of een PC-modem wil is dan met de **Sidecar** redelijk goedkoop uit. Ook zijn de expansion-boxes en harde schijven voor Amiga-DOS aanzienlijk duurder dan de PC-analoge machines.



Geheel anders is de situatie bij de IBM compatibele **Amiga 2000**.

Voor deze 68000-8088/80286 MS-DOS-hybride zijn tal van redelijk geprijsde PC-uitbreidingen te koop en meerdere vrije slots beschikbaar. De kopers van de Amiga 2000 hebben ongetwijfeld bij de aankoop al enige MS-DOS-toepassingen in het achterhoofd. Bezitters van het oudere 1000-model en de 500-hobbycomputer zijn in deze aan duurdere speciale hardware-uitbreidingen gebonden.

De Amiga Transformer

Voor de softwarematige emulatie van de Intel 8088 4.77 MHz processor levert Commodore de Amiga **Transformer** van Simile Research. Het complete pakket bestaat uit een 3.5 inch diskette, het manual en wat plakkerijtjes voor het aanpassen van de toetsenbord-layout. Mocht u als Amiga-bezitter dit pakket niet standaard meegeleverd hebben gekregen dan is het voor rond de f 230,- los of in combinatie met een Amiga 5.25 inch floppy drive te koop.

De Amiga Transformer biedt de gebruiker de volgende faciliteiten:

- Compatibiliteit met de bestaande MS-DOS software
- 640 K aan vrij RAM
- Een seriële (COM1) en parallelle poort
- Het gebruik van 3.5 en 5.25 inch in het 40-sporen (360 K) formaat.
- Alle DOS- en BIOS-interrupts
- 80-koloms monochrome video-display
- Direct lezen en wegschrijven van/naar het videogeheugen
- Het gebruik van MS-DOS compatibele randapparatuur
- Een IBM PC XT toetsenbord-layout

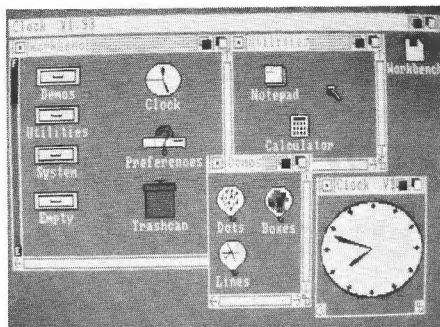
Het installeren van de Transformer is tamelijk bewerkelijk. Eerst dient de gebruiker een extra 256 K (dus in totaal 512 K) RAM-uitbreidingskaart in een poort te drukken. Vervolgens een extra diskdrive aan te sluiten. Bij voorkeur een 5.25 inch type daar 3.5 inch MS-DOS programmatuur slechts mondjesmaat verkrijgbaar is. Het manual geeft de schema's voor het aansluiten van de verschillende floppy drive combinaties. Maximaal kunnen drie externe diskdrives als een soort *daisy chain* worden aangesloten.

Menu's

Na (!) het aansluiten van de drives wordt de Amiga met randapparatuur normaal aangeschakeld (NB. de 5.25 inch drives voor de 3.5 inch modellen!) en opgestart. Na het laden van Workbench gaat een kopie van de Transformer in de drive en kan het ikoon met de muis geactiveerd worden. Het activeren van de Transformer resulteert in het verwijderen van alle Amiga-DOS files uit het RAM en het installeren van de software-emulator. Daarna vraagt het programma om substitutie door een MS-DOS schijf (versie 2.0 of hoger). Al met al heel wat bewerkelijker en tijdrovender dan het opstarten van een PC op de eigen besturingsschijf.

De Transformer disk kent een aantal **preference-menu's**. Deze zijn via het *Set ATP refs.-tool* toegankelijk voor wijzigingen. Het **Disk Menu** installeert de floppy drives onder elk een eigen letter. Op het **Video Menu** kan de ge-

bruiker de cursorvorm en de display modifieren (highlighted, onderstrepen, reverse e.d.). Het **Color Menu** doet bij de 1.0 versie nog niets. De monitor blijft groen. Bij toekomstige Transformer-versies komt daar verandering in. Het **Mono Menu** staat wel toe om behalve groen-zwart een andere kleuren combinaties in te stellen, bijvoorbeeld zwart-wit of amber-zwart.



De geheugen configuratie verloopt via het Memory Menu. Daarbij staan de volledige MS-DOS 640 K aan vrij RAM ter beschikking. Overigens biedt de Amiga zelf maximaal 8.5 MB aan vrij RAM. Via de functietoetsen wordt de gewenste hoeveelheid K-RAM ingesteld.

Tot slot is er nog een **Attachment Menu** dat het aanpassen van de keyboard layout en het gebruik van muis, digitizer, lichtpen of joystick regelt. Verder kan de gebruiker met dit menu de hoogte en hardheid van de geluidssignalen instellen.

Problemen

De meeste door ons geprobeerde MS-DOS pakketten draaiden met Transformer zonder mankeren op de Amiga. Problemen ontstonden door de traagheid van de emulator, circa de helft van de normale XT-snelheid, en de afwijkende keyboard layout. Met name memory residente en met hotkeys werkende software zorgde af en toe voor ongewenste effecten. Het werkte soms helemaal niet, slechts half of anders dan gepland. Mogelijk wordt dit in nieuwere releases verbeterd. Ook zullen de aan echte PC-toetsenborden gewende gebruikers er dikwijls naast slaan. Iemand die op de Amiga begint zal hier echter nauwelijks last van ondervinden.

Het Engelse manual is goed verzorgd. Wel vinden wij dat er veel waarschuwingen in staan. Een verkeerde handeling kan de gebruiker blijkbaar zuur opbreken en dat is toch wel een minpunt in het Amiga-ontwerp.

De nieuwe Amiga 1020 5.25 inch floppy drive werkt snel en vlekkeloos met

de bestaande MS-DOS diskettes. Vermoedelijk zal de winkelprijs van deze drive rond de f800,- komen te liggen.

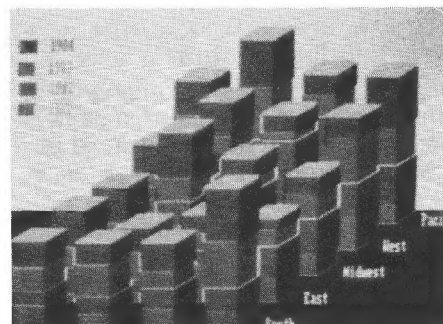
Een probleem vormt nog het minder ganbare 3.5 inch floppy format. Overspelen via een 5.25 inch drive of seriële uitgang is noodzakelijk om de MS-DOS programmatuur in de Amiga 500 of 1000 te laden.

Het werken met de Transformer heeft uiteraard alleen zin op de nieuwe 500- en oudere 1000-modellen. Het snelheidsverlies wordt daar voor lief genomen. Bij de 2000 is het draaien onder een transformer behalve een goedkope oplossing, pure verspilling van de geboden mogelijkheden.

Sidecar

Amiga's zijspan is een wat vormloze doos die op de rechter uitbreidingspoort van de Amiga wordt aangesloten. Qua stijl past de Sidecar niet zo bij de Amiga 500 en 1000 terwijl het geheel behoorlijk veel bureauruimte in beslag neemt. Bovendien vereist de gedraaide expansiepoort van de 500 ook nog een apart ontsierend aansluitkabeltje. Andere ontwerpers van Amiga-randapparatuur hebben dat wel beter opgelost. Die werken de hoogte in en dan past de combinatie systeemkast-expansionbox beter bij de monitor.

Ook geeft de constructie van de expansiepoort bij de Amiga 1000 te denken. Deze uitbreidingsbus vormt namelijk de enige connectie tussen de hoofdkaart en de Sidecar. De kans op beschadiging door geforceerd aandrukken of stoten lijkt ons allerm minst denkbeeldig. Wat listige verborgen schroefverbindingen tussen de systeemkast en Sidecar hadden meer soliditeit gegeven.



Met het aansluiten van de Sidecar is de gebruiker meteen de expansiebus kwijt. Die heeft men maar op de zijspan weggelaten. Gelukkig zitten er wel twee spelpoorten op de Sidecar. Als PC-kloon stelt de Sidecar niet veel

voor. Er zijn slechts een 360 K floppy drive, 256 KRAM, een 8088 4.77 MHz CPU, IBM-PC compatibele BIOS-ROMs en drie uitbreidings-slots (waarvan een voor de hard disk controller) aan boord. Het toetsenbord blijft het eigen Amiga keyboard. In combinatie met de Amiga, extra RAM en een harde schijf nemen de MS-DOS mogelijkheden echter sterk toe.

De harde schijf kan zowel door MS- als AmigaDos benut worden en de Amiga biedt zelf multitasking via een apart MS-DOS venster.

Het on-board RAM van de Sidecar is toegankelijk voor de Amiga zelf. Dat is voorlopig weer 512 K extra Amiga-RAM en binnenkort gaat Commodore zelfs een 2MB RAM Expansion kaart voor de Sidecar op de markt brengen. Liefhebbers van grote spreadsheets, gedetailleerde graphics en animaties kunnen dan hun hart ophalen.

Het systeem kan dus, in tegenstelling tot de Transformer, tegelijk Amiga- en MS-DOS programma's draaien.

De video-output verloopt via de eigen Amiga monitor. Doordat Sidecar de beeldinformatie rechtstreeks naar het 128K video-geheugen wegschrijft ontstaan er geen problemen met lastige programma's zoals Flight-simulator. Speciale bijgeleverde koppelingssoftware zorgt er voor dat de gewenste monochrome en/of kleur-schermen in de Amiga-vensters worden gemuleerd. Het gebruik van een EGA-kaart is helaas nog niet mogelijk.

De Sidecar kan verder de seriële en parallelle poorten van de Amiga aansturen, via een extra socket rekenen met 8087 co-processor en externe 3.5 of 5.25 inch drives lezen en schrijven. Zwakke punten blijven de snelheid en het niet geheel IBM PC-compatibele Amiga keyboard. De snelheid is gewoon 4.77 MHz (circa twee maal sneller dan de software-emulator) en daarmee aanzienlijk lager dan de kloksnelheid van de Motorola 68000, een AT of Turbo XT. Een tweede tijd-consumerende bezigheid is het opstarten. Na het laden van Sidekick en Workbench (bij het model 500 al in het ROM gebakken) volgt de koppelingssoftware. Al met al heel wat bewerkelijker dan het opstarten vanaf een MS-DOS schijf op een IBM PC-kloon. Over het keyboard hebben wij het al bij de Transformer gehad. Nog niet over de toch wel stevige prijs van f 2400,- (inclusief BTW). Ons inziens kan dit bedrag best wat lager.

XT- en AT-kaarten voor de 2000

Het **A2088 board** of Janus-kaart is een in feite een zelfstandige XT-hoofdkaart. Een Sidecar op een kaart dus. De 8088 microprocessor draait onder de standaard 4.77 Mhz klokcyclus en een 16KB EPROM bevat alle noodzakelijke BIOS-routines. Het BIOS is door Commodore behoorlijk aangepast om communicatie met de Amiga 68000-chip mogelijk te maken. De A2088 heeft standaard 512 K aan boord plus nog eens een extra 128K voor het dual-port RAM. Een SMC 9268 floppy controller chip en een standaard 5.25 inch floppy drive connector kunnen maximaal twee (een interne en een externe drive) daisy chained floppystations aansturen.

Om alle XT-functies op een enkele kaart te kunnen proppen moest Commodore eigen (custom)-chips ontwerpen. Een speciale **multifunction chip** functioneert daarbij zowel als een DMA- en interrupt controller als een timer voor alle bussignalen. Andere chips "bedriegen" het Sidecar-deel van de Amiga 2000. De machine "gelooft" dat hij een echte XT is en via het dual-port memory worden alle toetsaanslagen en I/O-opdrachten van de Amiga als zijnde een PC XT ge-emuleerd.

Verder biedt de XT-kaart een extra socket voor de 8087 mathematische co-processor. Aan de onderzijde van de A2088 PC emulator-kaart zitten twee grote connectoren. De grootste connector wordt in het Amiga bus-connectorslot gestoken. De kleinste in het PC-deel van de 2000.

De A2088 wordt natuurlijk ook softwarematig ondersteund vanuit Kickstart en de Intuition werkbank. Via interrupts trekt de ene computer de aandacht van de andere. Zo heeft de XT-kaart een eigen PC-ikoon (naar keuze monochroom of in kleur) op de werkbank waarin informatie en graphics opgetekend worden.

PCWindows kan ook, met maximaal 7 interrupts, onder multitasking draaien. Zo is het bijvoorbeeld mogelijk om dBASE op de XT te laten sorteren terwijl de Amiga zelf voor tekstverwerking of het maken van animaties vrij blijft.

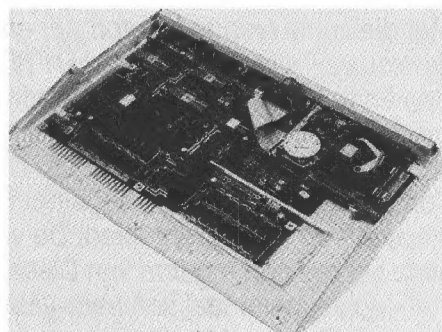
Stel de A2088 wil een grafiek op de Amiga-monitor tekenen. De PC-emulator zendt dan een interrupt naar de Amiga, vervolgens kijkt de 68000 naar het video display memory deel van het

dual-port RAM en gaat dan via een EXEC-sig-naal tot de grafische uitvoer over. Het programmaatje **PC-Windows** opent voor de XT een eigen venster met informatiescherm. De prijs van de A2088 Janus-kaart zal ongeveer rond de f 1400,- komen te liggen.

'Snelle kaart'

Gelijk met de XT-kaart werd een AT-insteekkaart aangekondigd. Wij konden er nog geen hand op leggen, maar volgens de specificaties bevat dit emulator board:

- ° Een 80286 CPU met een kloksnelheid van 8MHz
- ° 512KB (plus 128 dual-port) aan vrij RAM
- ° Een 16 KB BIOS-ROM
- ° Een floppy-controller voor en 5.25 1.2 MB en een 3.5 inch 720 KB drive
- ° Een socket voor de 80287 mathematische co-processor.



De prijs van dit **A2286 AT-board** wordt op iets van f 2200,- geraamd. Voor de toekomst denkt Commodore nog aan een 386 emulator, maar daarover staat niets vast.

Conclusie

Soft- en hardware MS-DOS emulatie blijft enigszins behelpen. Wie MS-DOS pakketten wil draaien of zijn Amiga via de Sidecar wil uitbreiden zal de beperkingen moeten slikken. Wie echt MS-DOS op de Amiga wil draaien en voor een nieuwe aankoop staat kan beter direct in de Amiga 2000 investeren. Voor de 500 en 1000 gaat onze voorkeur naar de mogelijkheden van de Sidecar ten opzichte van de veel goedkopere software emulator. Sidecar is een doordacht stuk randapparatuur dat echter nog wel voor verbetering vatbaar is.